

CEIT – 04 – 502A
EE04L / MWF / 4:30p – 6:00p
LABORATORY REPORT 2

GROUP 6

Group Members:

Bernardo, Raevon Thaddeus C.

Bertumen, Charles Jefferson

Cabanes, Christine Joy P.

Cesar, John Lester M.

Landicho, Bhaves Nicolette D.

Solis, Johnloyd P.

Machine Problem 2: Root Approximation by Bracketing Method

Program Name: Group 6 Working Bisection Method Root Approximator

Acronym: G6-WBMRA

Current Version: 1.0.2

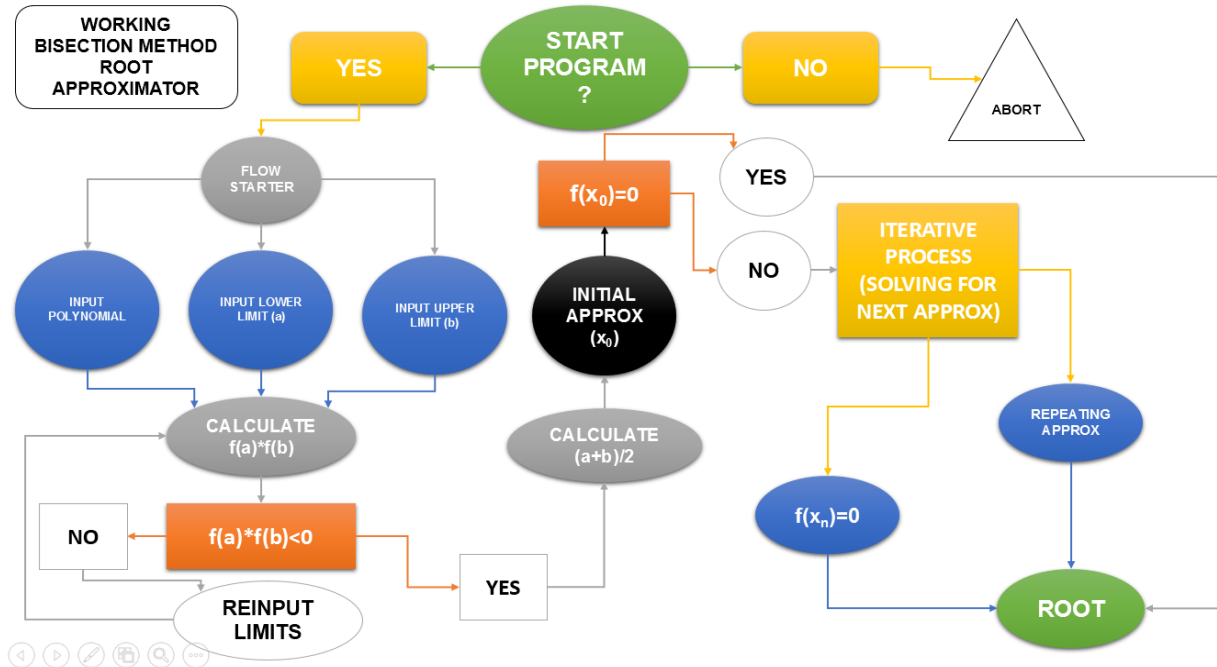
Version History:

- **1.0.1**
 - Trial version
 - Designed in Scilab version 6.0.2
 - Can only approximate up to 15th approximation
- **1.0.2**
 - Updated trial version
 - Designed in Scilab version 6.0.2
 - Can now approximate up to 20th approximation

TABLE OF CONTENTS

- I. Flow Chart
- II. Source Code of the Working Program
- III. Program Instructions
- IV. Sample Output
- V. Program Accuracy
- VI. Development Team Contributions

I. Flow Chart



The program starts with an option to proceed or abort. Upon agreeing to start the program, the user will be prompted by the program to input desired (1) *polynomial function*, (2) *lower limit or value of a*, and (3) *upper limit or value of b*. After user inputs desired function and values, the program will start to calculate until $f(a)$ and $f(b)$ are identified. The algorithm of the system itself will first verify if the product of $f(a)$ and $f(b)$ will be less than 0 before proceeding to the next command. However, if the system determines that the product is greater than or equal to 0, then the user will be prompted to input other values for a and b . Upon verifying that the product is actually less than 0, the program will then compute for the value of x_0 and will be required to reinput the value after being rounded off by the user to 3 decimal places. Afterwards, the system will compute for $f(x_0)$. If the value of $f(x_0)$ is determined to be equal to 0, then by the program's algorithm, x_0 or the initial approximation will be determined as the root of the polynomial function inputted by the user. On the other hand, if that is not the case, the program will proceed to calculate for the next approximation, following its designed algorithms. An iterative process will be conducted by the system with the values inputted by the user in each approximation until it reaches the final or n^{th} approximation wherein the root of the polynomial function is finally determined. It is also important to keep in mind that as of now, the current capabilities of the program version itself may automatically abort the entire system progress if the root is still not determined on the 20th approximation. This is due to the program having reached its current maximum capability of processing approximations and will therefore not exceed past the 20th approximation.

II. Source Code of the Working Program

```
clc
disp("Welcome to Group 6 Working Bisection Method Root Approximator 1.0.2 (G6-WBMRA 1.0.2)!")
disp("Version: 1.0.2 (UPDATED TRIAL VERSION)")
disp("Current Version Capabilities: Can now approximate up to x20 or twentieth approximation.")
disp("")
disp("How to Use G6-WBMRA 1.0.2: Input polynomial function in this format: poly([a,b,c,-->
nth],%x%,%coeff%)")
disp("")
disp("Wherein: Letters a,b,c,-->nth (excluding x) corresponds to the coefficients of your desired
polynomial function in an ascending order in terms of degree.")
disp("Wherein: The symbol % should be replaced with quotation marks (IMPORTANT)")
disp("e.g. poly([1,2,3],%x%,%coeff%) will input 1+2x+3x^2")
disp("")
disp("Further instructions will be given as you progress onwards.")
disp(" ")
YES=1
NO=0
ANSWER=input("Start the program? Input YES to proceed and NO to abort: ")
if ANSWER==1 then
    disp("PROGRAM START!!")
    disp("")
    y=input("Input polynomial function: ")
    a1=input("Input lower limit or value of a: ")
    b1=input("Input upper limit or value of b: ")
    disp("polynomial function: "+string(y)+"")
    disp("")
elseif ANSWER==0 then
    clc
    disp("You have aborted the program. Thank you for using G6-WBMRA 1.0.2!")
    abort
else
    clc
    disp("ERROR: Instructions not followed. Automated abort command.")
    abort
end
if a1>b1 then
    disp("ERROR: The value of a should not be higher than b.")
    disp("")
    disp("Please reinput lower and upper limits.")
    disp("")
    a1=input("Input lower limit or value of a: ")
    b1=input("Input upper limit or value of b: ")
end
if a1>b1 then
    disp("ERROR: The value of a should not be higher than b.")
    disp("")
    disp("Please reinput lower and upper limits.")
    disp("")
    a1=input("Input lower limit or value of a: ")
    b1=input("Input upper limit or value of b: ")
end
if a1>b1 then
    disp("ERROR: The value of a should not be higher than b.")
```

```

disp("")
disp("Please reinput lower and upper limits.")
disp("")
a1=input("Input lower limit or value of a: ")
b1=input("Input upper limit or value of b: ")
elseif a1>b1 then
    clc
    disp("ERROR: Instructions not followed. Automated abort command.")
    abort
end
horner(y,[a1])
a2=horner(y,[a1]) // a2=f(a)
disp("value of f(a): "+string(a2)+" ")
horner(y,[b1])
b2=horner(y,[b1]) // b2=f(b)
disp("value of f(b): "+string(b2)+" ")
c1=a2*b2 // c1=f(a)*f(b)
if c1<0 then
    d1=(a1+b1)/2 // d1=x0 or initial approximation
    disp("initial approximation: "+string(d1)+" ")
else
    disp("ERROR: f(a)*f(b)is not less than 0.")
    disp("")
    disp("Please reinput lower and upper limits.")
    disp("")
    a1=input("Input lower limit or value of a: ")
    b1=input("Input upper limit or value of b: ")
    horner(y,[a1])
    a2=horner(y,[a1]) // a2=f(a)
    disp("value of f(a): "+string(a2)+" ")
    horner(y,[b1])
    b2=horner(y,[b1]) // b2=f(b)
    disp("value of f(b): "+string(b2)+" ")
    c1=a2*b2 // c1=f(a)*f(b)
    if c1<0 then
        d1=(a1+b1)/2 // d1=x0 or initial approximation
        disp("initial approximation: "+string(d1)+" ")
    else
        disp("ERROR: f(a)*f(b)is not less than 0.")
        disp("")
        disp("Please reinput lower and upper limits.")
        disp("")
        a1=input("Input lower limit or value of a: ")
        b1=input("Input upper limit or value of b: ")
        horner(y,[a1])
        a2=horner(y,[a1]) // a2=f(a)
        disp("value of f(a): "+string(a2)+" ")
        horner(y,[b1])
        b2=horner(y,[b1]) // b2=f(b)
        disp("value of f(b): "+string(b2)+" ")
        c1=a2*b2 // c1=f(a)*f(b)
        if c1<0 then
            d1=(a1+b1)/2 // d1=x0 or initial approximation
            disp("initial approximation: "+string(d1)+" ")
        else
            disp("ERROR: f(a)*f(b)is not less than 0.")

```

```

disp("")
disp("Please reinput lower and upper limits.")
disp("")
a1=input ("Input lower limit or value of a: ")
b1=input ("Input upper limit or value of b: ")
horner(y,[a1])
a2=horner(y,[a1]) // a2=f(a)
disp("value of f(a): "+string(a2)+" ")
horner(y,[b1])
b2=horner(y,[b1]) // b2=f(b)
disp("value of f(b): "+string(b2)+" ")
c1=a2*b2 // c1=f(a)*f(b)
if c1<0 then
    d1=(a1+b1)/2 // d1=x0 or initial approximation
    disp("initial approximation: "+string(d1)+" ")
else
    clc
    disp("ERROR: Please Reboot Program to try again.")
    abort
end
end
end
disp("Please input initial approximation or x0 that is rounded-off to 3 decimal places including its zeroes.")
d2=input ("x0 or initial approximation: ")
disp("value of x0: "+string(d2)+" ") //d2=x0
horner(y,[d2])
d3=horner(y,[d2]) // d3=f(x0)
disp("value of f(x0): "+string(d3)+"")
if d3==0 then
    disp (" "+string(d2)+" is the root.")
    abort
end
// f(x0)*f(a) or f(x0)*f(b)
if (d3*a2)<0 then
    e1=(d2+a1)/2
    disp("first approximation: "+string(e1)+"")
elseif (d3*b2)<0 then
    e1=(d2+b1)/2
    disp("first approximation: "+string(e1)+"")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
// e1=x1
disp("Please input first approximation or x1 that is rounded-off to 3 decimal places including its zeroes.")
e2=input ("x1 or first approximation: ")
disp("value of x1: "+string(e2)+" ")
horner(y,[e2])
e3=horner(y,[e2]) // e3=f(x1)
disp("value of f(x1): "+string(e3)+" ")
if e3==0 then
    disp (" "+string(e2)+" is the root.")
    abort
end

```

```

end
if (e3*d3)<0 then
    f1=(e2+d2)/2
    disp("second approximation: "+string(f1)+" ")
elseif (e3*b2)<0 then
    f1=(e2+b1)/2
    disp("second approximation: "+string(f1)+" ")
elseif (e3*a2)<0 then
    f1=(e2+a1)/2
    disp("second approximation: "+string(f1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer for error encountered. Thank you!")
    disp("")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input second approximation or x2 that is rounded-off to 3 decimal places including its zeroes.")
f2=input("x2 or second approximation: ")
disp("value of x2: "+string(f2)+" ")
if f2==e2 then
    disp (" "+string(f2)+" is the root.")
    abort
else
    horner(y,[f2])
end
f3=horner(y,[f2]) // f3=f(x2)
disp("value of f(x2): "+string(f3)+" ")
if f3==0 then
    disp (" "+string(f2)+" is the root.")
    abort
end
if (f3*e3)<0 then
    g1=(f2+e2)/2
    disp("third approximation: "+string(g1)+" ")
elseif (f3*d3)<0 then
    g1=(f2+d2)/2
    disp("third approximation: "+string(g1)+" ")
elseif (f3*b2)<0 then
    g1=(f2+b1)/2
    disp("third approximation: "+string(g1)+" ")
elseif (f3*a2)<0 then
    g1=(f2+a1)/2
    disp("third approximation: "+string(g1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input third approximation or x3 that is rounded-off to 3 decimal places including its zeroes.")
g2=input("x3 or third approximation: ")
disp("value of x3: "+string(g2)+" ")
if g2==f2 then
    disp (" "+string(g2)+" is the root.")

```

```

    abort
elseif g2==e2 then
    disp (""+string(g2)+" is the root.")
    abort
elseif g2==d2 then
    disp (""+string(g2)+" is the root.")
    abort
else
    horner(y,[g2])
end
g3=horner(y,[g2]) // g3=f(x3)
disp("value of f(x3): "+string(g3)+" ")
if g3==0 then
    disp (""+string(g2)+" is the root.")
    abort
end
if (g3*f3)<0 then
    h1=(g2+f2)/2
    disp("fourth approximation: "+string(h1)+" ")
elseif (g3*e3)<0 then
    h1=(g2+e2)/2
    disp("fourth approximation: "+string(h1)+" ")
elseif (g3*d3)<0 then
    h1=(g2+d2)/2
    disp("fourth approximation: "+string(h1)+" ")
elseif (g3*b2)<0 then
    h1=(g2+b1)/2
    disp("fourth approximation: "+string(h1)+" ")
elseif (g3*a2)<0 then
    h1=(g2+a1)/2
    disp("fourth approximation: "+string(h1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input fourth approximation or x4 that is rounded-off to 3 decimal places including its zeroes.")
h2=input ("x4 or fourth approximation: ")
disp("value of x4: "+string(h2)+" ")
if h2==g2 then
    disp (""+string(h2)+" is the root.")
    abort
elseif h2==f2 then
    disp (""+string(h2)+" is the root.")
    abort
elseif h2==e2 then
    disp (""+string(h2)+" is the root.")
    abort
elseif h2==d2 then
    disp (""+string(h2)+" is the root.")
    abort
else
    horner(y,[h2])
end

```

```

h3=horner(y,[h2]) // h3=f(x4)
disp("value of f(x4): "+string(h3)+" ")
if h3==0 then
    disp (" "+string(h2)+" is the root.")
    abort
end
if (h3*g3)<0 then
    i1=(h2+g2)/2
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*f3)<0 then
    i1=(h2+f2)/2
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*e3)<0 then
    i1=(h2+e2)/2
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*d3)<0 then
    i1=(h2+d2)/2
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*b2)<0 then
    i1=(h2+b1)/2
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*a2)<0 then
    i1=(h2+a1)/2
    disp("fifth approximation: "+string(i1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input fifth approximation or x5 that is rounded-off to 3 decimal places including its zeroes.")
i2=input ("x5 or fifth approximation: ")
disp("value of x5: "+string(i2)+" ")
if i2==h2 then
    disp (" "+string(i2)+" is the root.")
    abort
elseif i2==g2 then
    disp (" "+string(i2)+" is the root.")
    abort
elseif i2==f2 then
    disp (" "+string(i2)+" is the root.")
    abort
elseif i2==e2 then
    disp (" "+string(i2)+" is the root.")
    abort
elseif i2==d2 then
    disp (" "+string(i2)+" is the root.")
    abort
else
    horner(y,[i2])
end
i3=horner(y,[i2]) // i3=f(x5)
disp("value of f(x5): "+string(i3)+" ")
if i3==0 then
    disp (" "+string(i2)+" is the root.")
    abort

```



```

end
if (i3*h3)<0 then
    j1=(i2+h2)/2
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*g3)<0 then
    j1=(i2+g2)/2
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*f3)<0 then
    j1=(i2+f2)/2
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*e3)<0 then
    j1=(i2+e2)/2
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*d3)<0 then
    j1=(i2+d2)/2
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*b2)<0 then
    j1=(i2+b1)/2
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*a2)<0 then
    j1=(i2+a1)/2
    disp("sixth approximation: "+string(j1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input sixth approximation or x6 that is rounded-off to 3 decimal places including its zeroes.")
j2=input("x6 or sixth approximation: ")
disp("value of x6: "+string(j2)+" ")
if j2==i2 then
    disp (" "+string(j2)+" is the root.")
    abort
elseif j2==h2 then
    disp (" "+string(j2)+" is the root.")
    abort
elseif j2==g2 then
    disp (" "+string(j2)+" is the root.")
    abort
elseif j2==f2 then
    disp (" "+string(j2)+" is the root.")
    abort
elseif j2==e2 then
    disp (" "+string(j2)+" is the root.")
    abort
elseif j2==d2 then
    disp (" "+string(j2)+" is the root.")
    abort
else
    horner(y,[j2])
end
j3=horner(y,[j2]) // j3=f(x6)
disp("value of f(x6): "+string(j3)+" ")
if j3==0 then
    disp (" "+string(j2)+" is the root.")

```

```

    abort
end
if (j3*i3)<0 then
    k1=(j2+i2)/2
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*h3)<0 then
    k1=(j2+h2)/2
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*g3)<0 then
    k1=(j2+g2)/2
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*f3)<0 then
    k1=(j2+f2)/2
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*e3)<0 then
    k1=(j2+e2)/2
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*d3)<0 then
    k1=(j2+d2)/2
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*b2)<0 then
    k1=(j2+b1)/2
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*a2)<0 then
    k1=(j2+a1)/2
    disp("seventh approximation: "+string(k1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input seventh approximation or x7 that is rounded-off to 3 decimal places including its zeroes.")
k2=input("x7 or seventh approximation: ")
disp("value of x7: "+string(k2)+" ")
if k2==j2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==i2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==h2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==g2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==f2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==e2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==d2 then
    disp (" "+string(k2)+" is the root.")

```

```

    abort
else
    horner(y,[k2])
end
k3=horner(y,[k2]) // k3=f(x7)
disp("value of f(x7): "+string(k3)+" ")
if k3==0 then
    disp (" "+string(k2)+" is the root.")
    abort
end
if (k3*j3)<0 then
    l1=(j2+j2)/2
    disp("eighth approximation: "+string(l1)+" ")
elseif (k3*i3)<0 then
    l1=(j2+i2)/2
    disp("eighth approximation: "+string(l1)+" ")
elseif (k3*h3)<0 then
    l1=(j2+h2)/2
    disp("eighth approximation: "+string(l1)+" ")
elseif (k3*g3)<0 then
    l1=(j2+g2)/2
    disp("eighth approximation: "+string(l1)+" ")
elseif (k3*f3)<0 then
    l1=(j2+f2)/2
    disp("eighth approximation: "+string(l1)+" ")
elseif (k3*e3)<0 then
    l1=(j2+e2)/2
    disp("eighth approximation: "+string(l1)+" ")
elseif (k3*d3)<0 then
    l1=(j2+d2)/2
    disp("eighth approximation: "+string(l1)+" ")
elseif (k3*b2)<0 then
    l1=(j2+b1)/2
    disp("eighth approximation: "+string(l1)+" ")
elseif (k3*a2)<0 then
    l1=(j2+a1)/2
    disp("eighth approximation: "+string(l1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input eighth approximation or x8 that is rounded-off to 3 decimal places including its zeroes.")
l2=input ("x8 or eighth approximation: ")
disp("value of x8: "+string(l2)+" ")
if l2==k2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==j2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==i2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==h2 then

```

```

    disp (""+string(l2)+" is the root.")
    abort
elseif l2==g2 then
    disp (""+string(l2)+" is the root.")
    abort
elseif l2==f2 then
    disp (""+string(l2)+" is the root.")
    abort
elseif l2==e2 then
    disp (""+string(l2)+" is the root.")
    abort
elseif l2==d2 then
    disp (""+string(l2)+" is the root.")
    abort
else
    horner(y,[l2])
end
l3=horner(y,[l2]) // l3=f(x8)
disp("value of f(x8): "+string(l3)+" ")
if l3==0 then
    disp (""+string(l2)+" is the root.")
    abort
end
if (l3*k3)<0 then
    m1=(l2+k2)/2
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*j3)<0 then
    m1=(l2+j2)/2
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*i3)<0 then
    m1=(l2+i2)/2
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*h3)<0 then
    m1=(l2+h2)/2
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*g3)<0 then
    m1=(l2+g2)/2
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*f3)<0 then
    m1=(l2+f2)/2
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*e3)<0 then
    m1=(l2+e2)/2
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*d3)<0 then
    m1=(l2+d2)/2
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*b2)<0 then
    m1=(l2+b1)/2
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*a2)<0 then
    m1=(l2+a1)/2
    disp("ninth approximation: "+string(m1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")

```

```

disp("Contact Email: bernardoraevon@gmail.com")
abort
end
disp("Please input ninth approximation or x9 that is rounded-off to 3 decimal places including its zeroes.")
m2=input ("x9 or ninth approximation: ")
disp("value of x9: "+string(m2)+" ")
if m2==l2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==k2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==j2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==i2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==h2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==g2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==f2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==e2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==d2 then
    disp (" "+string(m2)+" is the root.")
    abort
else
    horner(y,[m2])
end
m3=horner(y,[m2]) // m3=f(x9)
disp("value of f(x9): "+string(m3)+" ")
if m3==0 then
    disp (" "+string(m2)+" is the root.")
    abort
end
if (m3*l3)<0 then
    n1=(m2+l2)/2
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*k3)<0 then
    n1=(m2+k2)/2
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*j3)<0 then
    n1=(m2+j2)/2
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*i3)<0 then
    n1=(m2+i2)/2
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*h3)<0 then
    n1=(m2+h2)/2

```

```

    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*g3)<0 then
    n1=(m2+g2)/2
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*f3)<0 then
    n1=(m2+f2)/2
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*e3)<0 then
    n1=(m2+e2)/2
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*d3)<0 then
    n1=(m2+d2)/2
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*b2)<0 then
    n1=(m2+b1)/2
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*a2)<0 then
    n1=(m2+a1)/2
    disp("tenth approximation: "+string(n1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input tenth approximation or x10 that is rounded-off to 3 decimal places including its zeroes.")
n2=input ("x10 or tenth approximation: ")
disp("value of x10: "+string(n2)+" ")
if n2==m2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==l2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==k2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==j2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==i2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==h2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==g2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==f2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==e2 then
    disp (" "+string(n2)+" is the root.")
    abort

```

```

elseif n2==d2 then
    disp (" "+string(n2)+" is the root.")
    abort
else
    horner(y,[n2])
end
n3=horner(y,[n2]) // n3=f(x10)
disp("value of f(x10): "+string(n3)+" ")
if n3==0 then
    disp (" "+string(n2)+" is the root.")
    abort
end
if (n3*m3)<0 then
    o1=(n2+m2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*l3)<0 then
    o1=(n2+l2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*k3)<0 then
    o1=(n2+k2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*j3)<0 then
    o1=(n2+j2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*i3)<0 then
    o1=(n2+i2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*h3)<0 then
    o1=(n2+h2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*g3)<0 then
    o1=(n2+g2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*f3)<0 then
    o1=(n2+f2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*e3)<0 then
    o1=(n2+e2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*d3)<0 then
    o1=(n2+d2)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*b2)<0 then
    o1=(n2+b1)/2
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*a2)<0 then
    o1=(n2+a1)/2
    disp("eleventh approximation: "+string(o1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input eleventh approximation or x11 that is rounded-off to 3 decimal places including its zeroes.")

```

```

o2=input("x11 or eleventh approximation: ")
disp("value of x11: "+string(o2)+" ")
if o2==n2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==m2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==l2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==k2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==j2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==i2 then
    disp(" "+string(n2)+" is the root.")
    abort
elseif o2==h2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==g2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==f2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==e2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==d2 then
    disp(" "+string(o2)+" is the root.")
    abort
else
    horner(y,[o2])
end
o3=horner(y,[o2]) // o3=f(x11)
disp("value of f(x11): "+string(o3)+" ")
if o3==0 then
    disp(" "+string(o2)+" is the root.")
    abort
end
if (o3*n3)<0 then
    p1=(o2+n2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*m3)<0 then
    p1=(o2+m2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*l3)<0 then
    p1=(o2+l2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*k3)<0 then
    p1=(o2+k2)/2
    disp("twelfth approximation: "+string(p1)+" ")

```



```

elseif (o3*j3)<0 then
    p1=(o2+j2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*i3)<0 then
    p1=(o2+i2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*h3)<0 then
    p1=(o2+h2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*g3)<0 then
    p1=(o2+g2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*f3)<0 then
    p1=(o2+f2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*e3)<0 then
    p1=(o2+e2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*d3)<0 then
    p1=(o2+d2)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*b2)<0 then
    p1=(o2+b1)/2
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*a2)<0 then
    p1=(o2+a1)/2
    disp("twelfth approximation: "+string(p1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input twelfth approximation or x12 that is rounded-off to 3 decimal places including its zeroes.")
p2=input("x12 or twelfth approximation: ")
disp("value of x12: "+string(p2)+" ")
if p2==o2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==n2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==m2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==l2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==k2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==j2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==i2 then

```

```

disp (" "+string(p2)+" is the root.")
abort
elseif p2==h2 then
disp (" "+string(p2)+" is the root.")
abort
elseif p2==g2 then
disp (" "+string(p2)+" is the root.")
abort
elseif p2==f2 then
disp (" "+string(p2)+" is the root.")
abort
elseif p2==e2 then
disp (" "+string(p2)+" is the root.")
abort
elseif p2==d2 then
disp (" "+string(p2)+" is the root.")
abort
else
horner(y,[p2])
end
p3=horner(y,[p2]) // p3=f(x12)
disp("value of f(x12): "+string(p3)+" ")
if p3==0 then
disp (" "+string(p2)+" is the root.")
abort
end
if (p3*o3)<0 then
q1=(p2+o2)/2
disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*n3)<0 then
q1=(p2+n2)/2
disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*m3)<0 then
q1=(p2+m2)/2
disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*l3)<0 then
q1=(p2+l2)/2
disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*k3)<0 then
q1=(p2+k2)/2
disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*j3)<0 then
q1=(p2+j2)/2
disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*i3)<0 then
q1=(p2+i2)/2
disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*h3)<0 then
q1=(p2+h2)/2
disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*g3)<0 then
q1=(p2+g2)/2
disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*f3)<0 then
q1=(p2+f2)/2
disp("thirteenth approximation: "+string(q1)+" ")

```

```

elseif (p3*e3)<0 then
    q1=(p2+e2)/2
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*d3)<0 then
    q1=(p2+d2)/2
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*b2)<0 then
    q1=(p2+b1)/2
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*a2)<0 then
    q1=(p2+a1)/2
    disp("thirteenth approximation: "+string(q1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input thirteenth approximation or x13 that is rounded-off to 3 decimal places including its zeroes.")
q2=input ("x13 or thirteenth approximation: ")
disp("value of x13: "+string(q2)+" ")
if q2==p2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==o2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==n2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==m2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==l2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==k2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==j2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==i2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==h2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==g2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==f2 then
    disp (" "+string(q2)+" is the root.")
    abort
elseif q2==e2 then

```

```

    disp (""+string(q2)+" is the root.")
    abort
elseif q2==d2 then
    disp (""+string(q2)+" is the root.")
    abort
else
    horner(y,[q2])
end
q3=horner(y,[q2]) // q3=f(x13)
disp("value of f(x13): "+string(q3)+" ")
if q3==0 then
    disp (""+string(q2)+" is the root.")
    abort
end
if (q3*p3)<0 then
    r1=(q2+p2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*o3)<0 then
    r1=(q2+o2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*n3)<0 then
    r1=(q2+n2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*m3)<0 then
    r1=(q2+m2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*l3)<0 then
    r1=(q2+l2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*k3)<0 then
    r1=(q2+k2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*j3)<0 then
    r1=(q2+j2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*i3)<0 then
    r1=(q2+i2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*h3)<0 then
    r1=(q2+h2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*g3)<0 then
    r1=(q2+g2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*f3)<0 then
    r1=(q2+f2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*e3)<0 then
    r1=(q2+e2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*d3)<0 then
    r1=(q2+d2)/2
    disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*b2)<0 then
    r1=(q2+b1)/2
    disp("fourteenth approximation: "+string(r1)+" ")

```

```

elseif (q3*a2)<0 then
    r1=(q2+a1)/2
    disp("fourteenth approximation: "+string(r1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input fourteenth approximation or x14 that is rounded-off to 3 decimal places including its zeroes.")
r2=input ("x14 or fourteenth approximation: ")
disp("value of x14: "+string(r2)+" ")
if r2==q2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==p2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==o2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==n2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==m2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==l2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==k2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==j2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==i2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==h2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==g2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==f2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==e2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==d2 then
    disp (" "+string(r2)+" is the root.")
    abort
else

```

```

    horner(y,[r2])
end
r3=horner(y,[r2]) // r3=f(x14)
disp("value of f(x14): "+string(r3)+" ")
if r3==0 then
    disp (" "+string(r2)+" is the root.")
    abort
end
if (r3*q3)<0 then
    s1=(r2+q2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*p3)<0 then
    s1=(r2+p2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*o3)<0 then
    s1=(r2+o2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*n3)<0 then
    s1=(r2+n2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*m3)<0 then
    s1=(r2+m2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*l3)<0 then
    s1=(r2+l2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*k3)<0 then
    s1=(r2+k2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*j3)<0 then
    s1=(r2+j2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*i3)<0 then
    s1=(r2+i2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*h3)<0 then
    s1=(r2+h2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*g3)<0 then
    s1=(r2+g2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*f3)<0 then
    s1=(r2+f2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*e3)<0 then
    s1=(r2+e2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*d3)<0 then
    s1=(r2+d2)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*b2)<0 then
    s1=(r2+b1)/2
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*a2)<0 then
    s1=(r2+a1)/2
    disp("fifteenth approximation: "+string(s1)+" ")

```

```

else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input fifteenth approximation or x15 that is rounded-off to 3 decimal places including its zeroes.")
s2=input ("x15 or fifteenth approximation: ")
disp("value of x15: "+string(s2)+" ")
if s2==r2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==q2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==p2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==o2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==n2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==m2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==l2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==k2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==j2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==i2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==h2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==g2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==f2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==e2 then
    disp (" "+string(s2)+" is the root.")
    abort
elseif s2==d2 then
    disp (" "+string(s2)+" is the root.")
    abort
else

```

```

    horner(y,[s2])
end
s3=horner(y,[s2]) // s3=f(x15)
disp("value of f(x15): "+string(s3)+" ")
if s3==0 then
    disp (" "+string(s2)+" is the root.")
    abort
end
if (s3*r3)<0 then
    t1=(s2+r2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*q3)<0 then
    t1=(s2+q2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*p3)<0 then
    t1=(s2+p2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*o3)<0 then
    t1=(s2+o2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*n3)<0 then
    t1=(s2+n2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*m3)<0 then
    t1=(s2+m2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*l3)<0 then
    t1=(s2+l2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*k3)<0 then
    t1=(s2+k2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*j3)<0 then
    t1=(s2+j2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*i3)<0 then
    t1=(s2+i2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*h3)<0 then
    t1=(s2+h2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*g3)<0 then
    t1=(s2+g2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*f3)<0 then
    t1=(s2+f2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*e3)<0 then
    t1=(s2+e2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*d3)<0 then
    t1=(s2+d2)/2
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*b3)<0 then
    t1=(s2+b1)/2
    disp("sixteenth approximation: "+string(t1)+" ")

```



```

elseif (s3*a2)<0 then
    t1=(s2+a1)/2
    disp("sixteenth approximation: "+string(t1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input sixteenth approximation or x16 that is rounded-off to 3 decimal places including its zeroes.")
t2=input("x16 or sixteenth approximation: ")
disp("value of x16: "+string(t2)+" ")
if t2==s2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==r2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==q2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==p2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==o2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==n2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==m2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==l2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==k2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==j2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==i2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==h2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==g2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==f2 then
    disp(" "+string(t2)+" is the root.")
    abort
elseif t2==e2 then

```

```

    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==d2 then
    disp (" "+string(t2)+" is the root.")
    abort
else
    horner(y,[t2])
end
t3=horner(y,[t2]) // t3=f(x16)
disp("value of f(x16): "+string(t3)+" ")
if t3==0 then
    disp (" "+string(t2)+" is the root.")
    abort
end
if (t3*s3)<0 then
    u1=(t2+s2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*r3)<0 then
    u1=(t2+r2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*q3)<0 then
    u1=(t2+q2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*p3)<0 then
    u1=(t2+p2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*o3)<0 then
    u1=(t2+o2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*n3)<0 then
    u1=(t2+n2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*m3)<0 then
    u1=(t2+m2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*l3)<0 then
    u1=(t2+l2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*k3)<0 then
    u1=(t2+k2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*j3)<0 then
    u1=(t2+j2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*i3)<0 then
    u1=(t2+i2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*h3)<0 then
    u1=(t2+h2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*g3)<0 then
    u1=(t2+g2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*f3)<0 then
    u1=(t2+f2)/2
    disp("seventeenth approximation: "+string(u1)+" ")

```

```

elseif (t3*e3)<0 then
    u1=(t2+e2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*d3)<0 then
    u1=(t2+d2)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*b2)<0 then
    u1=(t2+b1)/2
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*a2)<0 then
    u1=(t2+a1)/2
    disp("seventeenth approximation: "+string(u1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input seventeenth approximation or x17 that is rounded-off to 3 decimal places including its zeroes.")
u2=input ("x17 or seventeenth approximation: ")
disp("value of x17: "+string(u2)+" ")
if u2==t2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==s2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==r2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==q2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==p2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==o2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==n2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==m2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==l2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==k2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==j2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==i2 then

```

```

    disp (""+string(u2)+" is the root.")
    abort
elseif u2==h2 then
    disp (""+string(u2)+" is the root.")
    abort
elseif u2==g2 then
    disp (""+string(u2)+" is the root.")
    abort
elseif u2==f2 then
    disp (""+string(u2)+" is the root.")
    abort
elseif u2==e2 then
    disp (""+string(u2)+" is the root.")
    abort
elseif u2==d2 then
    disp (""+string(u2)+" is the root.")
    abort
else
    horner(y,[u2])
end
u3=horner(y,[u2]) // u3=f(x17)
disp("value of f(x17): "+string(u3)+" ")
if u3==0 then
    disp (""+string(u2)+" is the root.")
    abort
end
if (u3*t3)<0 then
    v1=(u2+t2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*s3)<0 then
    v1=(u2+s2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*r3)<0 then
    v1=(u2+r2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*q3)<0 then
    v1=(u2+q2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*p3)<0 then
    v1=(u2+p2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*o3)<0 then
    v1=(u2+o2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*n3)<0 then
    v1=(u2+n2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*m3)<0 then
    v1=(u2+m2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*l3)<0 then
    v1=(u2+l2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*k3)<0 then
    v1=(u2+k2)/2
    disp("eighteenth approximation: "+string(v1)+" ")

```

```

elseif (u3*j3)<0 then
    v1=(u2+j2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*i3)<0 then
    v1=(u2+i2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*h3)<0 then
    v1=(u2+h2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*g3)<0 then
    v1=(u2+g2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*f3)<0 then
    v1=(u2+f2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*e3)<0 then
    v1=(u2+e2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*d3)<0 then
    v1=(u2+d2)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*b2)<0 then
    v1=(u2+b1)/2
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*a2)<0 then
    v1=(u2+a1)/2
    disp("eighteenth approximation: "+string(v1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input eighteenth approximation or x18 that is rounded-off to 3 decimal places including its zeroes.")
v2=input("x18 or eighteenth approximation: ")
disp("value of x18: "+string(v2)+" ")
if v2==u2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==t2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==s2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==r2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==q2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==p2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==o2 then

```

```

    disp (""+string(v2)+" is the root.")
    abort
elseif v2==n2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==m2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==l2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==k2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==j2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==i2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==h2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==g2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==f2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==e2 then
    disp (""+string(v2)+" is the root.")
    abort
elseif v2==d2 then
    disp (""+string(v2)+" is the root.")
    abort
else
    horner(y,[v2])
end
v3=horner(y,[v2]) // v3=f(x18)
disp("value of f(x18): "+string(v3)+" ")
if v3==0 then
    disp (""+string(v2)+" is the root.")
    abort
end
if (v3*u3)<0 then
    w1=(v2+u2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*t3)<0 then
    w1=(v2+t2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*s3)<0 then
    w1=(v2+s2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*r3)<0 then
    w1=(v2+r2)/2
    disp("nineteenth approximation: "+string(w1)+" ")

```

```

elseif (v3*q3)<0 then
    w1=(v2+q2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*p3)<0 then
    w1=(v2+p2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*o3)<0 then
    w1=(v2+o2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*n3)<0 then
    w1=(v2+n2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*m3)<0 then
    w1=(v2+m2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*l3)<0 then
    w1=(v2+l2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*k3)<0 then
    w1=(v2+k2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*j3)<0 then
    w1=(v2+j2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*i3)<0 then
    w1=(v2+i2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*h3)<0 then
    w1=(v2+h2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*g3)<0 then
    w1=(v2+g2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*f3)<0 then
    w1=(v2+f2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*e3)<0 then
    w1=(v2+e2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*d3)<0 then
    w1=(v2+d2)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*b2)<0 then
    w1=(v2+b1)/2
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*a2)<0 then
    w1=(v2+a1)/2
    disp("nineteenth approximation: "+string(w1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input nineteenth approximation or x19 that is rounded-off to 3 decimal places including its zeroes.")

```

```

w2=input("x19 or nineteenth approximation: ")
disp("value of x19: "+string(w2)+" ")
if w2==v2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==u2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==t2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==s2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==r2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==q2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==p2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==o2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==n2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==m2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==l2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==k2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==j2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==i2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==h2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==g2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==f2 then
    disp(" "+string(w2)+" is the root.")
    abort
elseif w2==e2 then
    disp(" "+string(w2)+" is the root.")
    abort

```



```

elseif w2==d2 then
    disp (" "+string(w2)+" is the root.")
    abort
else
    horner(y,[w2])
end
w3=horner(y,[w2]) // w3=f(x19)
disp("value of f(x19): "+string(w3)+" ")
if w3==0 then
    disp (" "+string(w2)+" is the root.")
    abort
end
if (w3*v3)<0 then
    z1=(w2+v2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*u3)<0 then
    z1=(w2+u2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*t3)<0 then
    z1=(w2+t2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*s3)<0 then
    z1=(w2+s2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*r3)<0 then
    z1=(w2+r2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*q3)<0 then
    z1=(w2+q2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*p3)<0 then
    z1=(w2+p2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*o3)<0 then
    z1=(w2+o2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*n3)<0 then
    z1=(w2+n2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*m3)<0 then
    z1=(w2+m2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*l3)<0 then
    z1=(w2+l2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*k3)<0 then
    z1=(w2+k2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*j3)<0 then
    z1=(w2+j2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*i3)<0 then
    z1=(w2+i2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*h3)<0 then
    z1=(w2+h2)/2

```

```

    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*g3)<0 then
    z1=(w2+g2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*f3)<0 then
    z1=(w2+f2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*e3)<0 then
    z1=(w2+e2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*d3)<0 then
    z1=(w2+d2)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*b2)<0 then
    z1=(w2+b1)/2
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*a2)<0 then
    z1=(w2+a1)/2
    disp("twentieth approximation: "+string(z1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input twentieth approximation or x20 that is rounded-off to 3 decimal places including its zeroes.")
z2=input("x20 or twentieth approximation: ")
disp("value of x20: "+string(z2)+" ")
if z2==w2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==v2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==u2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==t2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==s2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==r2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==q2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==p2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==o2 then
    disp(" "+string(z2)+" is the root.")
    abort

```

```

elseif z2==n2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==m2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==l2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==k2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==j2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==i2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==h2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==g2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==f2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==e2 then
    disp (""+string(z2)+" is the root.")
    abort
elseif z2==d2 then
    disp (""+string(z2)+" is the root.")
    abort
else
    horner(y,[z2])
end
z3=horner(y,[z2]) // z3=f(x20)
disp("value of f(x20): "+string(z3)+" ")
if z3==0 then
    disp (""+string(z2)+" is the root.")
    abort
else
    disp("ERROR: Exceeded the capabilities of the program!")
    abort
end

```

III. Program Instructions

How to Use G6-WBMRA 1.0.2:

Input polynomial function in this format:

`poly([a,b,c,-->nth],%x%,%coeff%)`

Wherein: Letters a,b,c,-->nth (excluding x) corresponds to the coefficients of your desired polynomial function in an ascending order in terms of degree.

Wherein: The symbol % should be replaced with quotation marks (**IMPORTANT**)

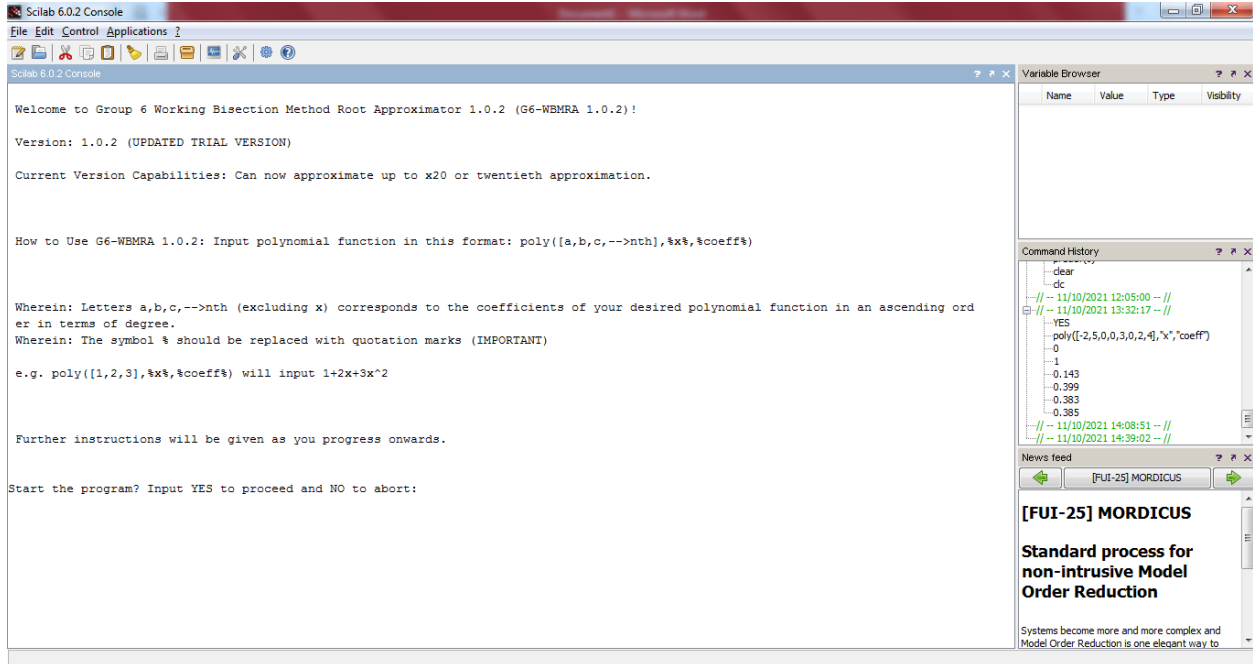
e.g. `poly([1,2,3],%x%,%coeff%)` will input **1+2x+3x²**

(Note: Due to the mechanics of Scilab, % is displayed in the instructions instead of quotation marks to avoid conflicts within the program codes.)

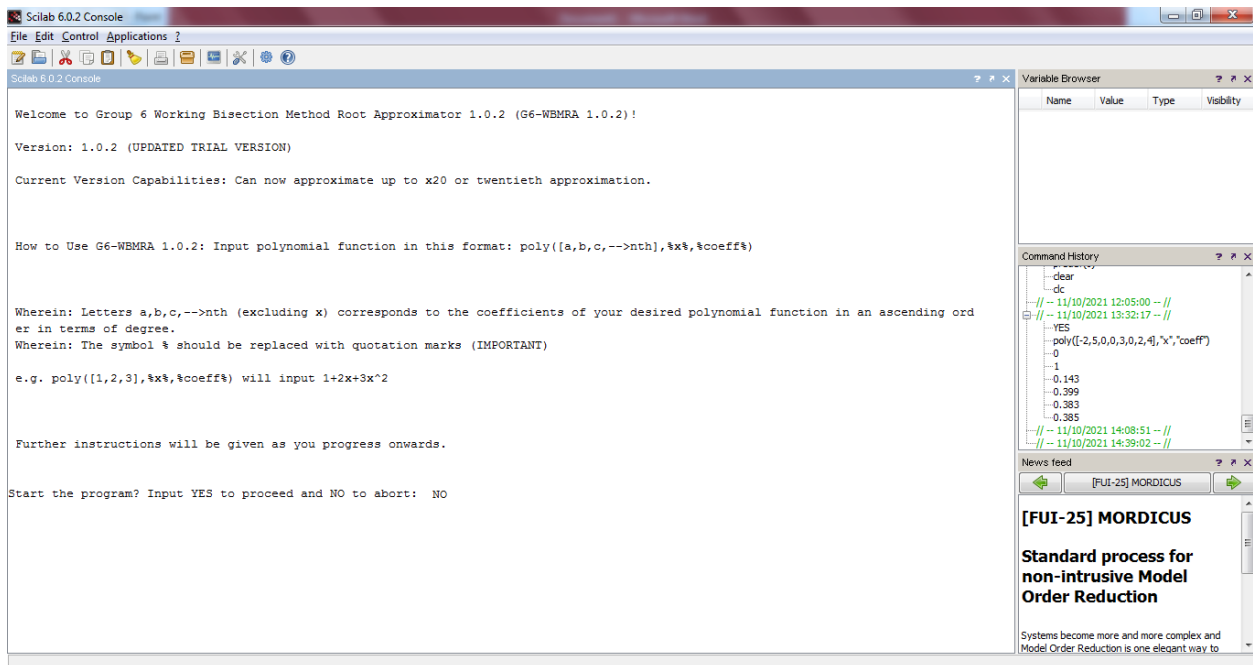
Further instructions will be given as you progress onwards.

IV. Sample Output

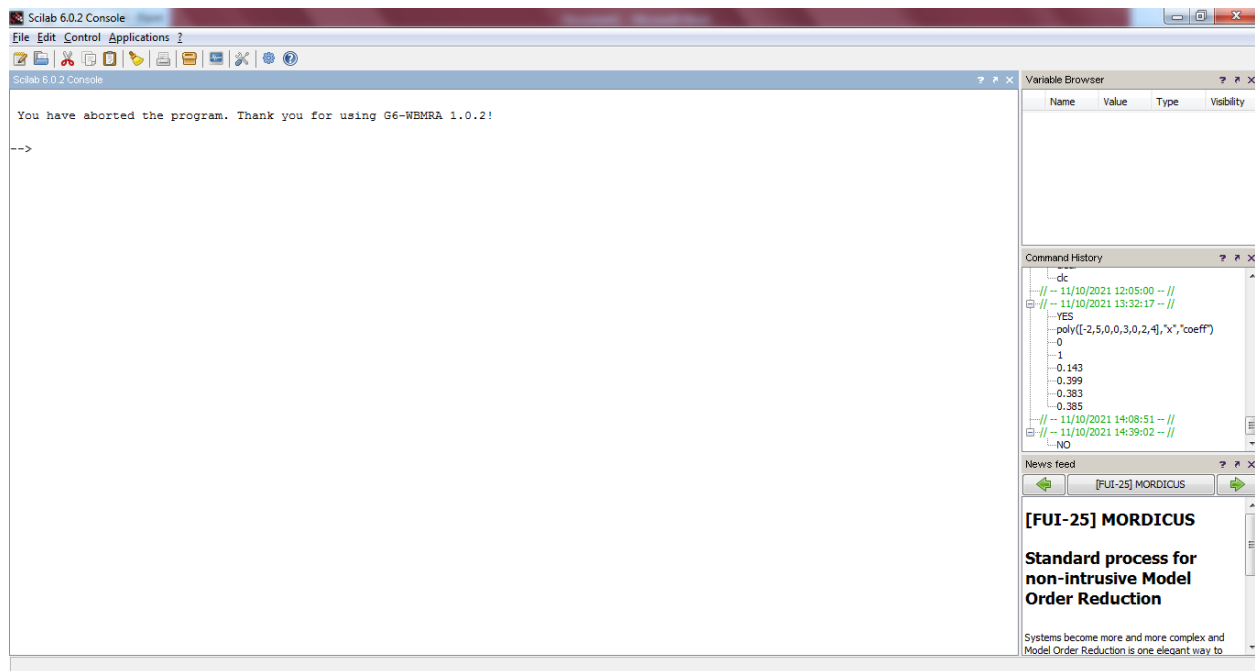
Program opened. Brief description and introduction of the program's current version as well as the set of instructions to be followed by the user will be displayed. The user will then be prompted for confirmation to either start or abort the program.



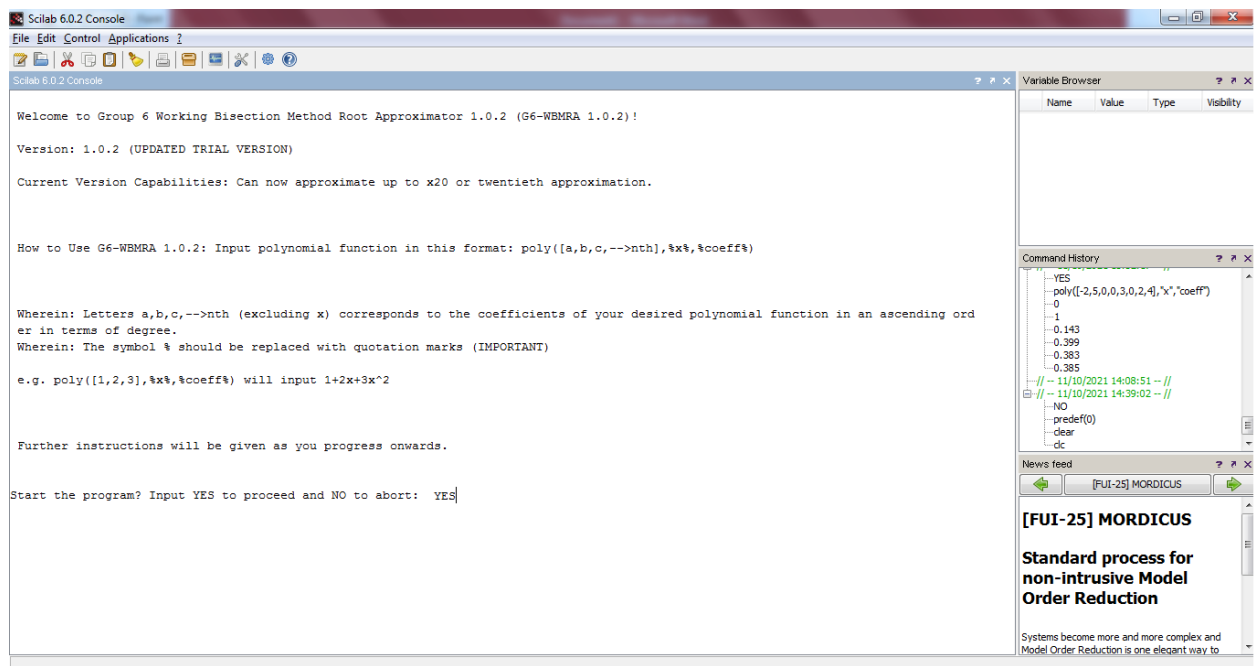
Scenario 1: User inputs "NO".



This is what will happen if the user's answer is "NO".



Scenario 2: User inputs "YES".



Scenario 1: After inputting “YES”, the user will be prompted to input desired polynomial function as well as the values of **a** and **b**. In this scenario, the algorithm has identified an error because the user had inputted a wrong assumption of lower and upper limits. As a result, the program will then prompt the user to input another different pair of values for the limits.

```

Scilab 6.0.2 Console
File Edit Control Applications ?

Start the program? Input YES to proceed and NO to abort: YES

PROGRAM START!!

Input polynomial function: poly([-3,2,5], "x", "coeff")
Input lower limit or value of a: -3
Input upper limit or value of b: -2

!polynomial function:      2      !
!                          !      !
!polynomial function:      - 3 + 2x + 5x      !

lower limit or a: -3
upper limit or b: -2
value of f(a): 36
value of f(b): 13

ERROR: f(a)*f(b) is not less than 0.

Please reinput lower and upper limits.

Input lower limit or value of a: |

```

Scenario 2: After inputting “YES”, the user will be prompted to input desired polynomial function as well as the values of **a** and **b**. In this scenario, the user has inputted a polynomial function as well as a right assumption of lower and upper limits. As a result, the system will calculate for the initial approximation and will determine if it is a root. If so, the program will end. If not, it will prompt the user to round off the value to 3 decimal places and reinput the said value to start calculation for the next approximation.

```

Scilab 6.0.2 Console
File Edit Control Applications ?

Further instructions will be given as you progress onwards.

Start the program? Input YES to proceed and NO to abort: YES

PROGRAM START!!

Input polynomial function: poly([-3,2,5], "x", "coeff")
Input lower limit or value of a: 0
Input upper limit or value of b: 1

!polynomial function:      2      !
!                          !      !
!polynomial function:      - 3 + 2x + 5x      !

lower limit or a: 0
upper limit or b: 1
value of f(a): -3
value of f(b): 4

initial approximation: 0.5

Please input initial approximation or x0 that is rounded-off to 3 decimal places including its zeroes.
x0 or initial approximation: |

```

After calculating the next approximation, the system will determine if it is a root. If not, the system will prompt the user to reinput the value of the approximation after rounding it off to 3 decimal places. This iterative process will continue until the root is determined. However, if the root is still not identified after the 20th approximation, the program will automatically abort as it is designed, in its current version, to only approximate until the 20th approximation.

```

Scilab 6.0.2 Console
File Edit Control Applications ?
Further instructions will be given as you progress onwards.
Start the program? Input YES to proceed and NO to abort: YES

PROGRAM START!!

Input polynomial function:      poly([-3,2,5],"x","coeff")
Input lower limit or value of a: 0
Input upper limit or value of b: 1

!polynomial function:          2      !
!                               !
!polynomial function:      - 3 + 2x + 5x  !

lower limit or a: 0
upper limit or b: 1
value of f(a):      -3
value of f(b):      4

initial approximation: 0.5

Please input initial approximation or x0 that is rounded-off to 3 decimal places including its zeroes.
x0 or initial approximation: 0.500

```

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```

--predef(0)
--clear
--dc
--YES
--poly([-3,2,5],"x","coeff")
--3
--2
--predef(0)
--clear
--dc
--YES
--poly([-3,2,5],"x","coeff")
--0
--1

```

News feed

[FUI-25] MORDICUS

[FUI-25] MORDICUS

Standard process for non-intrusive Model Order Reduction

Systems become more and more complex and Model Order Reduction is one elegant way to

```

Scilab 6.0.2 Console
File Edit Control Applications ?
Input polynomial function:      poly([-3,2,5],"x","coeff")
Input lower limit or value of a: 0
Input upper limit or value of b: 1

!polynomial function:          2      !
!                               !
!polynomial function:      - 3 + 2x + 5x  !

lower limit or a: 0
upper limit or b: 1
value of f(a):      -3
value of f(b):      4

initial approximation: 0.5

Please input initial approximation or x0 that is rounded-off to 3 decimal places including its zeroes.
x0 or initial approximation: 0.500

value of x0: 0.5
value of f(x0): -0.75
first approximation: 0.75

Please input first approximation or x1 that is rounded-off to 3 decimal places including its zeroes.
x1 or first approximation: 0.750

```

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```

--clear
--dc
--YES
--poly([-3,2,5],"x","coeff")
--3
--2
--predef(0)
--clear
--dc
--YES
--poly([-3,2,5],"x","coeff")
--0
--1
--0.500

```

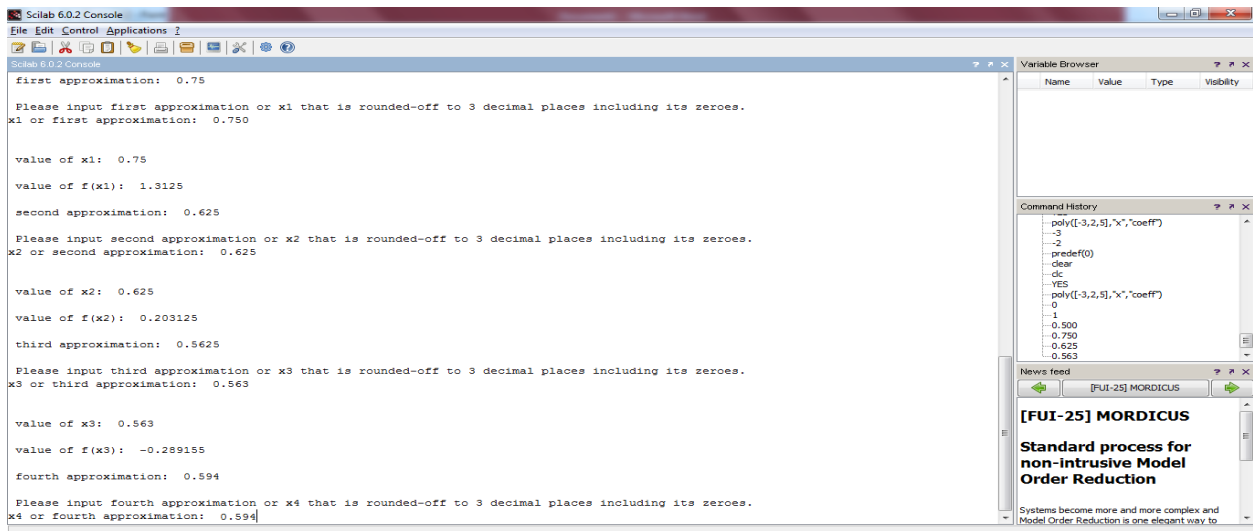
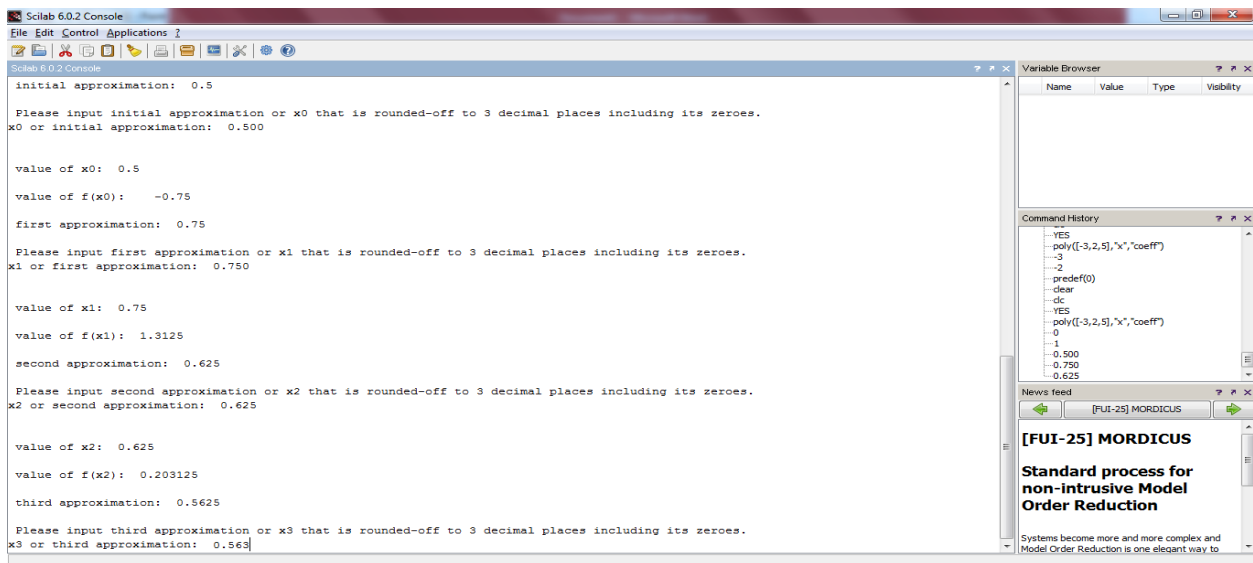
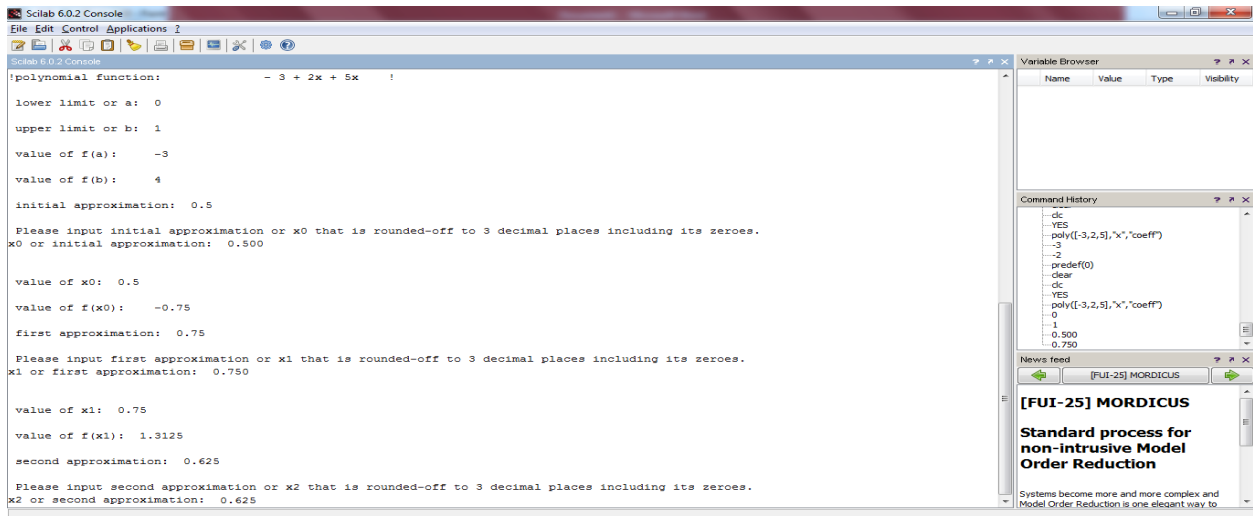
News feed

[FUI-25] MORDICUS

[FUI-25] MORDICUS

Standard process for non-intrusive Model Order Reduction

Systems become more and more complex and Model Order Reduction is one elegant way to



Scilab 6.0.2 Console

File Edit Control Applications ?

Scilab 6.0.2 Console

```
second approximation: 0.625
```

Please input second approximation or x2 that is rounded-off to 3 decimal places including its zeroes.
x2 or second approximation: 0.625

value of x2: 0.625
value of f(x2): 0.203125
third approximation: 0.5625

Please input third approximation or x3 that is rounded-off to 3 decimal places including its zeroes.
x3 or third approximation: 0.563

value of x3: 0.563
value of f(x3): -0.289155
fourth approximation: 0.594

Please input fourth approximation or x4 that is rounded-off to 3 decimal places including its zeroes.
x4 or fourth approximation: 0.594

value of x4: 0.594
value of f(x4): -0.04782
fifth approximation: 0.6095

Please input fifth approximation or x5 that is rounded-off to 3 decimal places including its zeroes.
x5 or fifth approximation: 0.610

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```
--3
--2
predef(0)
clear
clc
YES
poly([-3,2,5], 'x', 'coeff')
0
1
0.500
0.750
0.625
0.563
0.594
```

News feed

[FUI-25] MORDICUS

[FUI-25] MORDICUS

Standard process for non-intrusive Model Order Reduction

Systems become more and more complex and Model Order Reduction is one elegant way to

Scilab 6.0.2 Console

File Edit Control Applications ?

Scilab 6.0.2 Console

```
third approximation: 0.5625
```

Please input third approximation or x3 that is rounded-off to 3 decimal places including its zeroes.
x3 or third approximation: 0.563

value of x3: 0.563
value of f(x3): -0.289155
fourth approximation: 0.594

Please input fourth approximation or x4 that is rounded-off to 3 decimal places including its zeroes.
x4 or fourth approximation: 0.594

value of x4: 0.594
value of f(x4): -0.04782
fifth approximation: 0.6095

Please input fifth approximation or x5 that is rounded-off to 3 decimal places including its zeroes.
x5 or fifth approximation: 0.610

value of x5: 0.61
value of f(x5): 0.0805
sixth approximation: 0.602

Please input sixth approximation or x6 that is rounded-off to 3 decimal places including its zeroes.
x6 or sixth approximation: 0.602

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```
--2
predef(0)
clear
clc
YES
poly([-3,2,5], 'x', 'coeff')
0
1
0.500
0.750
0.625
0.563
0.594
0.610
```

News feed

[FUI-25] MORDICUS

[FUI-25] MORDICUS

Standard process for non-intrusive Model Order Reduction

Systems become more and more complex and Model Order Reduction is one elegant way to

Scilab 6.0.2 Console

File Edit Control Applications ?

Scilab 6.0.2 Console

```
fourth approximation: 0.594
```

Please input fourth approximation or x4 that is rounded-off to 3 decimal places including its zeroes.
x4 or fourth approximation: 0.594

value of x4: 0.594
value of f(x4): -0.04782
fifth approximation: 0.6095

Please input fifth approximation or x5 that is rounded-off to 3 decimal places including its zeroes.
x5 or fifth approximation: 0.610

value of x5: 0.61
value of f(x5): 0.0805
sixth approximation: 0.602

Please input sixth approximation or x6 that is rounded-off to 3 decimal places including its zeroes.
x6 or sixth approximation: 0.602

value of x6: 0.602
value of f(x6): 0.01602
seventh approximation: 0.598

Please input seventh approximation or x7 that is rounded-off to 3 decimal places including its zeroes.
x7 or seventh approximation: 0.598

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```
--predef(0)
--clear
--clc
--YES
--poly([-3,2,5], 'x', 'coeff')
--0
--1
--0.500
--0.750
--0.625
--0.563
--0.594
--0.610
--0.602
```

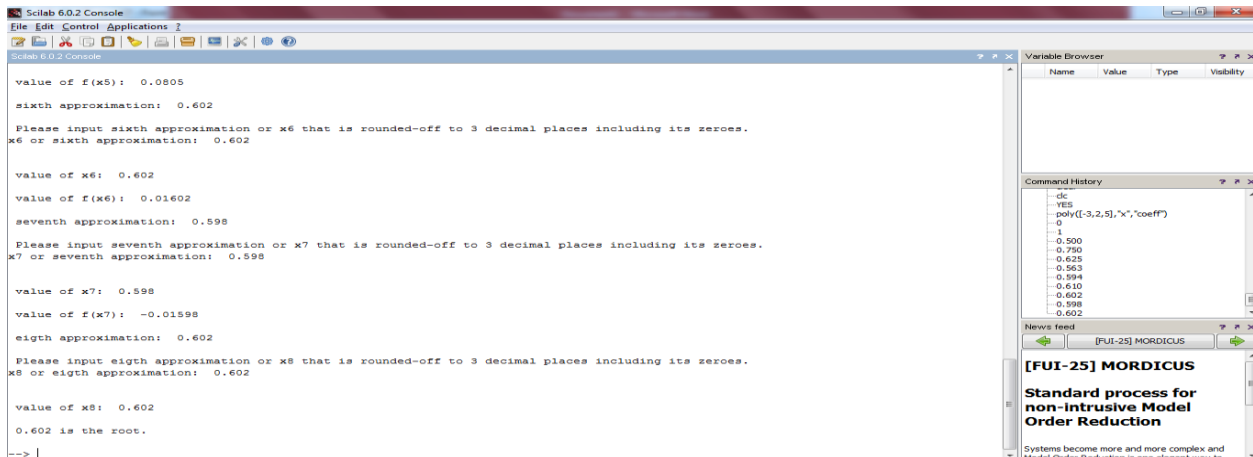
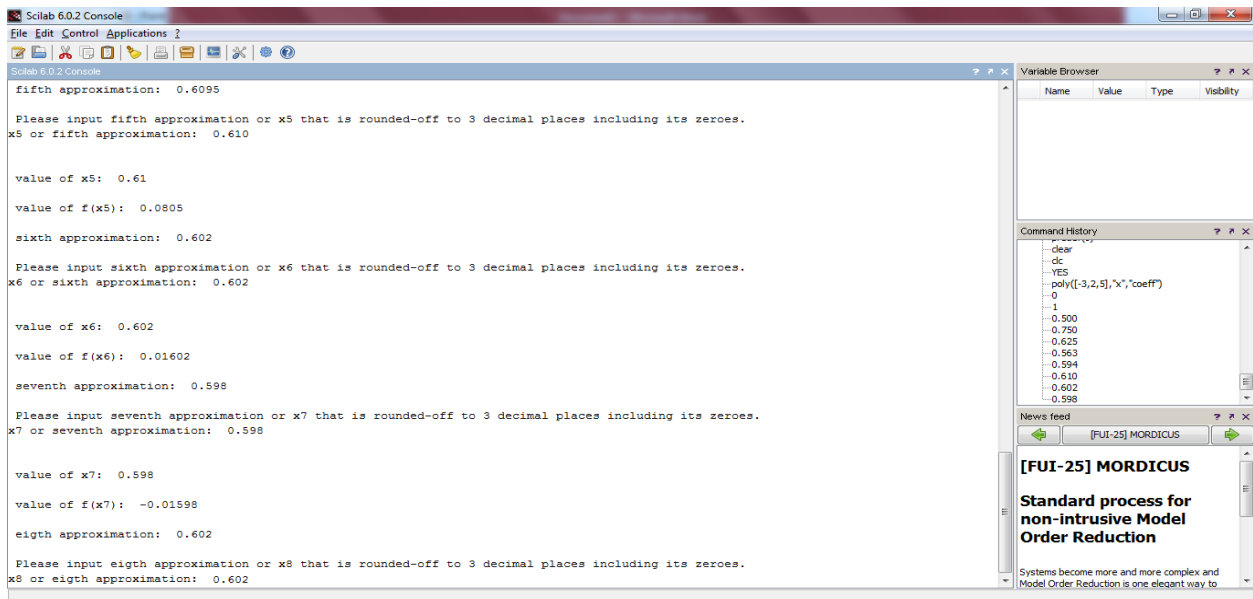
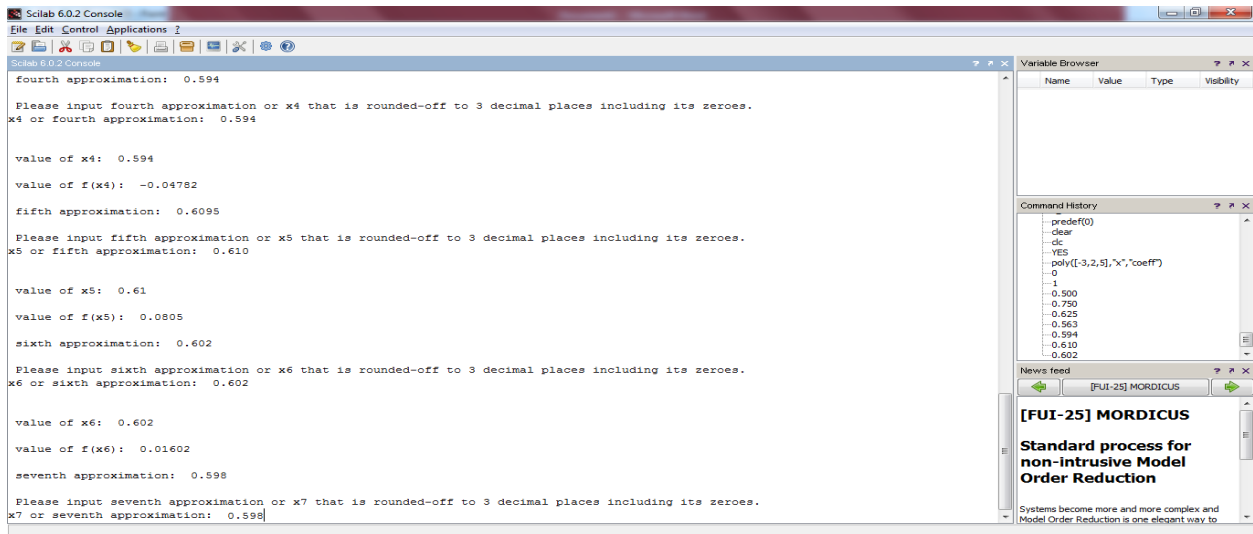
News feed

[FUI-25] MORDICUS

[FUI-25] MORDICUS

Standard process for non-intrusive Model Order Reduction

Systems become more and more complex and Model Order Reduction is one elegant way to



V. Program Accuracy

Accuracy Test #1

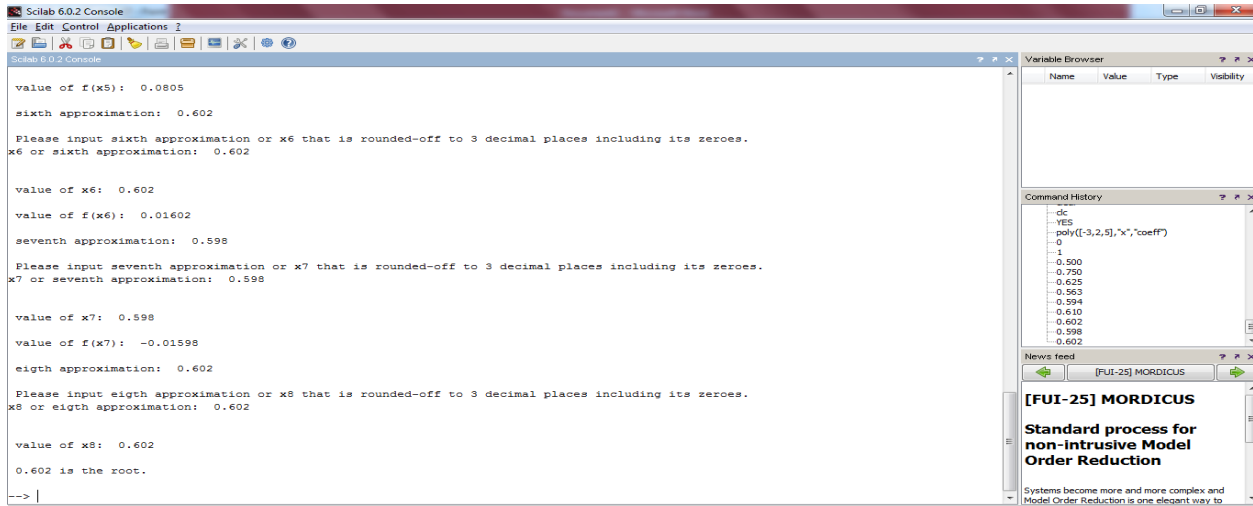
Polynomial function: $f(x) = 5x^2 + 2x - 3$

Lower limit or value of (a): 0

Upper limit or value of (b): 1

G6-WBMRA 1.0.2 Results:

Root = 0.602



```
Scilab 6.0.2 Console
File Edit Control Applications 2

value of f(x5): 0.0805
sixth approximation: 0.602

Please input sixth approximation or x6 that is rounded-off to 3 decimal places including its zeroes.
x6 or sixth approximation: 0.602

value of x6: 0.602
value of f(x6): 0.01602
seventh approximation: 0.598

Please input seventh approximation or x7 that is rounded-off to 3 decimal places including its zeroes.
x7 or seventh approximation: 0.598

value of x7: 0.598
value of f(x7): -0.01598
eighth approximation: 0.602

Please input eighth approximation or x8 that is rounded-off to 3 decimal places including its zeroes.
x8 or eighth approximation: 0.602

value of x8: 0.602
0.602 is the root.
--> |
```

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```
dc
YES
poly([-3,2,5], 'x', 'coeff')
-0
-1
-0.500
-0.750
-0.625
-0.563
-0.594
-0.610
-0.602
-0.598
-0.602
```

News feed

[FUI-25] MORDICUS

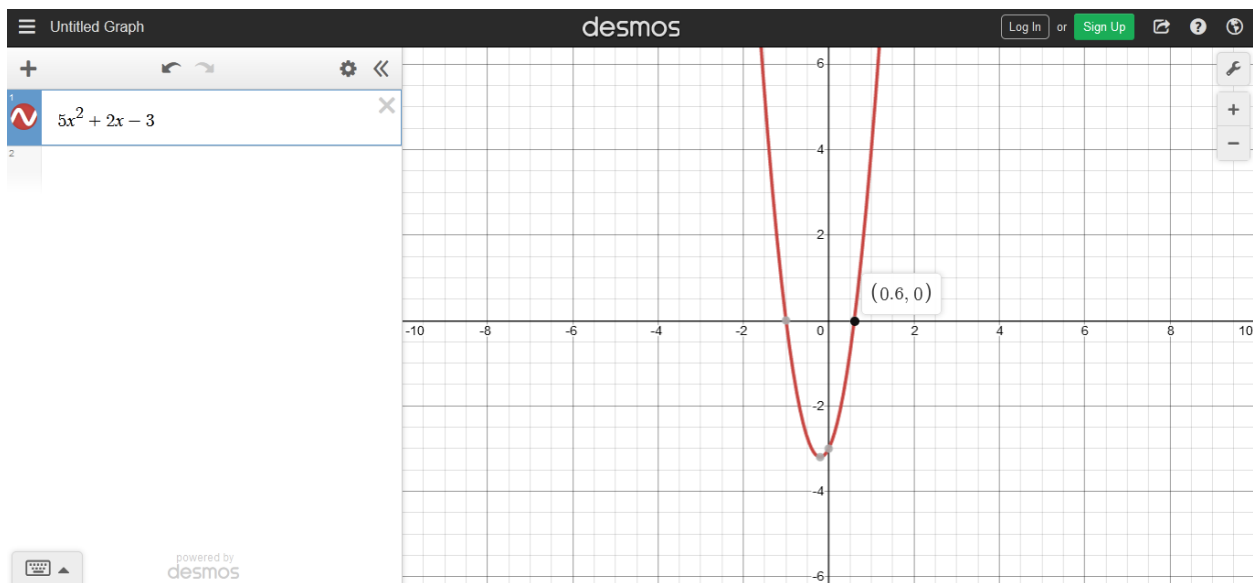
[FUI-25] MORDICUS

Standard process for non-intrusive Model Order Reduction

Systems become more and more complex and Model Order Reduction is one elegant way to

Desmos Graphing Calculator Results:

Root = 0.600



Accuracy Test #2

Polynomial function: $f(x) = 5x^5 + 3x^4 - 4x^3 - 2$

Lower limit or value of (a): 0

Upper limit or value of (b): 1

G6-WBMRA 1.0.2 Results:

Root = 0.900

```
Scilab 6.0.2 Console
File Edit Control Applications ?
Scilab 6.0.2 Console
value of f(x5): -0.1309002
sixth approximation: 0.899
Please input sixth approximation or x6 that is rounded-off to 3 decimal places including its zeroes.
x6 or sixth approximation: 0.900
value of x6: 0.9
value of f(x6): 0.00475
seventh approximation: 0.8955
Please input seventh approximation or x7 that is rounded-off to 3 decimal places including its zeroes.
x7 or seventh approximation: 0.896
value of x7: 0.896
value of f(x7): -0.0563313
eighth approximation: 0.9
Please input eighth approximation or x8 that is rounded-off to 3 decimal places including its zeroes.
x8 or eighth approximation: 0.900
value of x8: 0.9
0.9 is the root.
--> |
```

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```
dc
YES
poly([-2,0,0,-4,3,5], 'x', 'coeff')
0
1
0.500
0.750
0.875
0.938
0.907
0.891
0.900
0.896
0.900
```

News feed

[FUI-25] MORDICUS

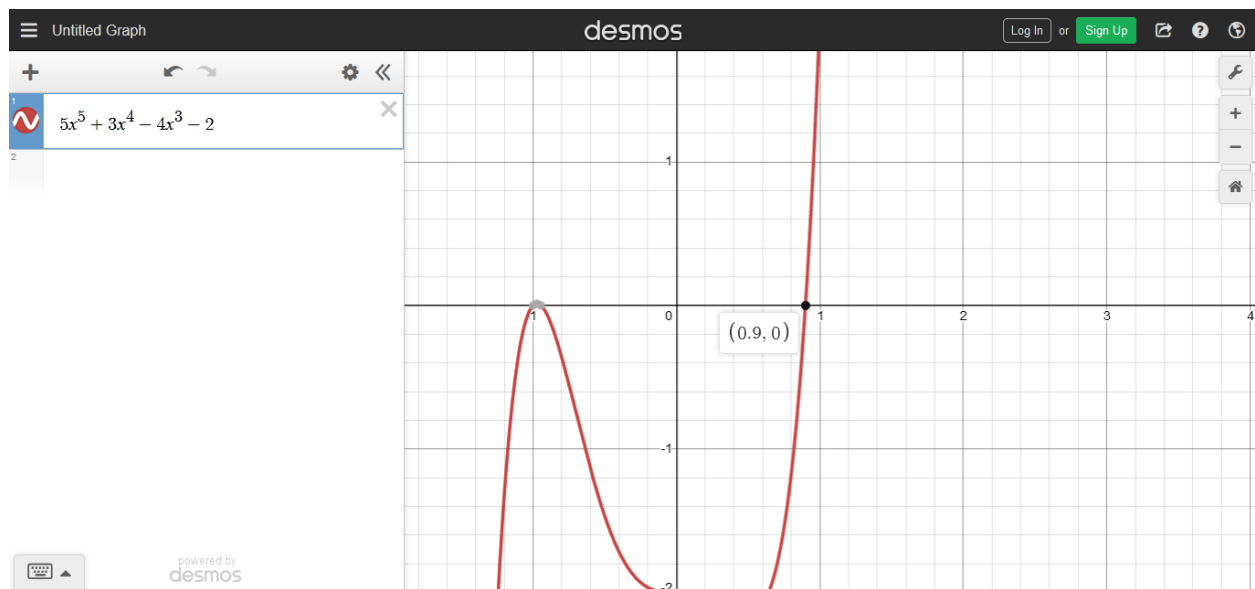
[FUI-25] MORDICUS

Standard process for non-intrusive Model Order Reduction

Systems become more and more complex and Model Order Reduction is one elegant way to

Desmos Graphing Calculator Results:

Root = 0.900



VI. Development Team Contributions

Development Team Members:

- ❖ Bernardo, Raevon Thaddeus C.
 - Head Developer & Programmer
 - Designed the algorithms of the working program
 - Final debugger of the program
- ❖ Bertumen, Charles Jefferson
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program
- ❖ Cabanes, Christine Joy P.
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program
- ❖ Cesar, John Lester M.
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program
- ❖ Landicho, Bhaves Nicolette D.
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program
- ❖ Solis, Johnloyd P.
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program

The development of the program was conducted systematically in order to maximize work efficiency, therefore, the final output was the result of total team effort and cooperation.

Head Developer's Remark

CEIT – 04 – 502A
EE04L / MWF / 4:30p – 6:00p
LABORATORY REPORT 2

GROUP 6

Group Members:

Bernardo, Raevon Thaddeus C.

Bertumen, Charles Jefferson

Cabanes, Christine Joy P.

Cesar, John Lester M.

Landicho, Bhaves Nicolette D.

Solis, Johnloyd P.

Machine Problem 2: Root Approximation by Bracketing Method

Program Name: Group 6 Working False Position Method Root Approximator

Acronym: G6–WFPMRA

Current Version: 1.0.2

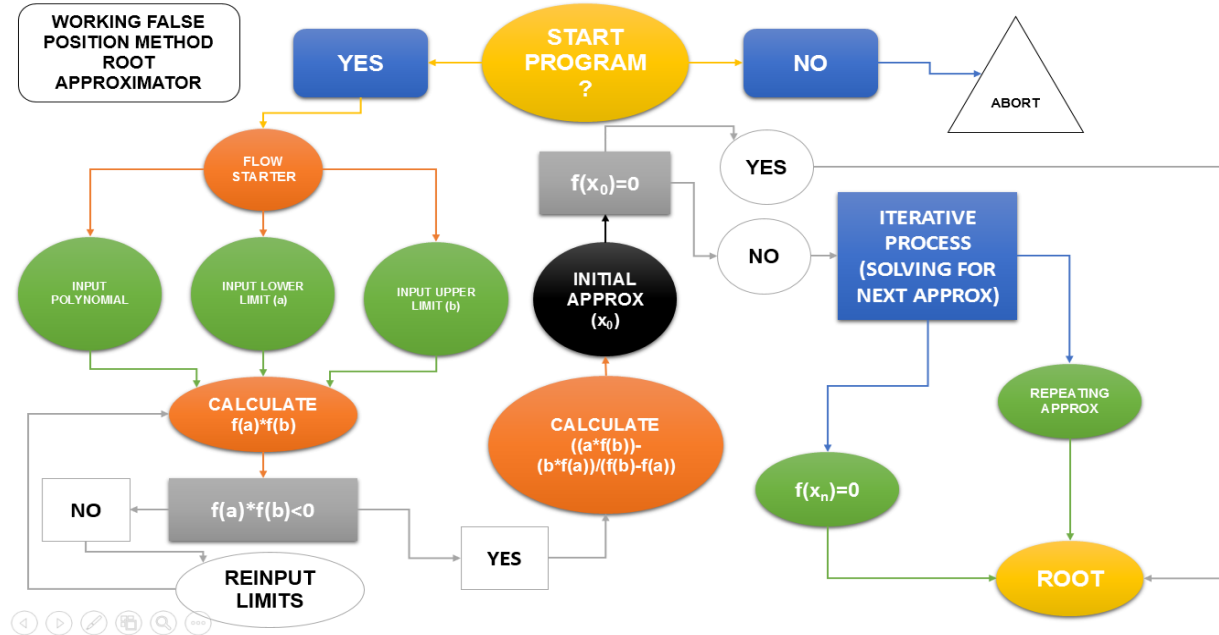
Version History:

- **1.0.1**
 - Trial version
 - Designed in Scilab version 6.0.2
 - Can only approximate up to 15th approximation
- **1.0.2**
 - Updated trial version
 - Designed in Scilab version 6.0.2
 - Can now approximate up to 20th approximation

TABLE OF CONTENTS

- I. Flow Chart
- II. Source Code of the Working Program
- III. Program Instructions
- IV. Sample Output
- V. Program Accuracy
- VI. Development Team Contributions

I. Flow Chart



The program starts with an option to proceed or abort. Upon agreeing to start the program, the user will be prompted by the program to input desired (1) *polynomial function*, (2) *lower limit or value of a*, and (3) *upper limit or value of b*. After user inputs desired function and values, the program will start to calculate until $f(a)$ and $f(b)$ are identified. The algorithm of the system itself will first verify if the product of $f(a)$ and $f(b)$ will be less than 0 before proceeding to the next command. However, if the system determines that the product is greater than or equal to 0, then the user will be prompted to input other values for a and b . Upon verifying that the product is actually less than 0, the program will then compute for the value of x_0 and the user will be required to reinput the value after being rounded off by the user to 3 decimal places. Afterwards, the system will compute for $f(x_0)$. If the value of $f(x_0)$ is determined to be equal to 0, then by the program's algorithm, x_0 or the initial approximation will be determined as the root of the polynomial function inputted by the user. On the other hand, if that is not the case, the user will be instructed to round the value off the displayed $f(x_0)$ to 3 decimal places and will be prompted to reinput the said value. The program will then proceed to calculate for the next approximation, following its designed algorithms. An iterative process will be conducted by the system with the values inputted by the user in each approximation until it reaches the final or n^{th} approximation wherein the root of the polynomial function is finally determined. It is also important to keep in mind that as of now, the current capabilities of the program version itself may automatically abort the entire system progress if the root is still not determined on the 20th approximation. This is due to the program having reached its current maximum capability of processing approximations and will therefore not exceed past the 20th approximation.

II. Source Code of the Working Program

```
clc
disp("Welcome to Group 6 Working False Position Method Root Approximator 1.0.2 (G6-WFPMRA 1.0.2)!")
disp("Version: 1.0.2 (UPDATED TRIAL VERSION)")
disp("Current Version Capabilities: Can now approximate up to x20 or twentieth approximation.")
disp("")
disp("How to Use G6-WFPMRA 1.0.2: Input polynomial function in this format: poly([a,b,c,-->nth],%x%,%coeff%)")
disp("")
disp("Wherein: Letters a,b,c,-->nth (excluding x) corresponds to the coefficients of your desired polynomial function in an ascending order in terms of degree.")
disp("Wherein: The symbol % should be replaced with quotation marks (IMPORTANT)")
disp("e.g. poly([1,2,3],%x%,%coeff%) will input 1+2x+3x^2")
disp("")
disp("Further instructions will be given as you progress onwards.")
disp(" ")
YES=1
NO=0
ANSWER=input("Start the program? Input YES to proceed and NO to abort: ")
if ANSWER==1 then
    disp("PROGRAM START!!")
    disp("")
    y=input("Input polynomial function: ")
    a1=input("Input lower limit or value of a: ")
    b1=input("Input upper limit or value of b: ")
    disp("polynomial function: "+string(y)+"")
    disp("")
elseif ANSWER==0 then
    clc
    disp("You have aborted the program. Thank you for using G6-WFPMRA 1.0.2!")
    abort
else
    clc
    disp("ERROR: Instructions not followed. Automated abort command.")
    abort
end
if a1>b1 then
    disp("ERROR: The value of a should not be higher than b.")
    disp("")
    disp("Please reinput lower and upper limits.")
    disp("")
    a1=input("Input lower limit or value of a: ")
    b1=input("Input upper limit or value of b: ")
end
if a1>b1 then
    disp("ERROR: The value of a should not be higher than b.")
    disp("")
    disp("Please reinput lower and upper limits.")
    disp("")
    a1=input("Input lower limit or value of a: ")
    b1=input("Input upper limit or value of b: ")
elseif a1>b1 then
    clc
    disp("ERROR: Instructions not followed. Automated abort command.")
```

```

    abort
end
horner(y,[a1])
a2=horner(y,[a1]) // a2=f(a)
disp("value of f(a): " +string(a2)+" ")
horner(y,[b1])
b2=horner(y,[b1]) // b2=f(b)
disp("value of f(b): " +string(b2)+" ")
c1=a2*b2 // c1=f(a)*f(b)
if c1<0 then
    d1=((a1*b2)-(b1*a2))/(b2-a2) // d1=x0 or initial approximation
    disp("initial approximation: " +string(d1)+" ")
else
    disp("ERROR: f(a)*f(b)is not less than 0.")
    disp("")
    disp("Please reinput lower and upper limits.")
    disp("")
    a1=input ("Input lower limit or value of a: ")
    b1=input ("Input upper limit or value of b: ")
    horner(y,[a1])
    a2=horner(y,[a1]) // a2=f(a)
    disp("value of f(a): " +string(a2)+" ")
    horner(y,[b1])
    b2=horner(y,[b1]) // b2=f(b)
    disp("value of f(b): " +string(b2)+" ")
    c1=a2*b2 // c1=f(a)*f(b)
    if c1<0 then
        d1=((a1*b2)-(b1*a2))/(b2-a2) // d1=x0 or initial approximation
        disp("initial approximation: " +string(d1)+" ")
    else
        disp("ERROR: f(a)*f(b)is not less than 0.")
        disp("")
        disp("Please reinput lower and upper limits.")
        disp("")
        a1=input ("Input lower limit or value of a: ")
        b1=input ("Input upper limit or value of b: ")
        horner(y,[a1])
        a2=horner(y,[a1]) // a2=f(a)
        disp("value of f(a): " +string(a2)+" ")
        horner(y,[b1])
        b2=horner(y,[b1]) // b2=f(b)
        disp("value of f(b): " +string(b2)+" ")
        c1=a2*b2 // c1=f(a)*f(b)
        if c1<0 then
            d1=((a1*b2)-(b1*a2))/(b2-a2) // d1=x0 or initial approximation
            disp("initial approximation: " +string(d1)+" ")
        else
            disp("ERROR: f(a)*f(b)is not less than 0.")
            disp("")
            disp("Please reinput lower and upper limits.")
            disp("")
            a1=input ("Input lower limit or value of a: ")
            b1=input ("Input upper limit or value of b: ")
            horner(y,[a1])
            a2=horner(y,[a1]) // a2=f(a)
            disp("value of f(a): " +string(a2)+" ")

```

```

    horner(y,[b1])
    b2=horner(y,[b1]) // b2=f(b)
    disp("value of f(b): "+string(b2)+" ")
    c1=a2*b2 // c1=f(a)*f(b)
    if c1<0 then
        d1=((a1*b2)-(b1*a2))/(b2-a2) // d1=x0 or initial approximation
        disp("initial approximation: "+string(d1)+" ")
    else
        clc
        disp("ERROR: Please Reboot Program to try again.")
        abort
    end
end
end
end
disp("Please input initial approximation or x0 that is rounded-off to 3 decimal places including its zeroes.")
d2=input ("x0 or initial approximation: ")
disp("value of x0: "+string(d2)+" ") //d2=x0
horner(y,[d2])
d3=horner(y,[d2]) // d3=f(x0)
disp("value of f(x0): "+string(d3)+"")
if d3==0 then
    disp (" "+string(d2)+" is the root.")
    abort
elseif d3~=0 then
    disp("Please input f(x0)that is rounded-off to 3 decimal places including its zeroes.")
    d3=input ("value of f(x0): ")
end
// f(x0)*f(a) or f(x0)*f(b)
if (d3*a2)<0 then
    e1=((d2*b2)-(b1*d3))/(b2-d3)
    disp("first approximation: "+string(e1)+"")
elseif (d3*b2)<0 then
    e1=((d2*a2)-(a1*d3))/(a2-d3)
    disp("first approximation: "+string(e1)+"")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
// e1=x1
disp("Please input first approximation or x1 that is rounded-off to 3 decimal places including its zeroes.")
e2=input ("x1 or first approximation: ")
disp("value of x1: "+string(e2)+" ")
if e2==d2 then
    disp (" "+string(e2)+" is the root.")
    abort
else
    horner(y,[e2])
end
e3=horner(y,[e2]) // e3=f(x1)
disp("value of f(x1): "+string(e3)+" ")
if e3==0 then
    disp (" "+string(e2)+" is the root.")
    abort

```

```

elseif e3~=0 then
    disp("Please input f(x1)that is rounded-off to 3 decimal places including its zeroes.")
    e3=input ("value of f(x1): ")
end
if (e3*d3)<0 then
    f1=((e2*d3)-(d2*e3))/(d3-e3)
    disp("second approximation: "+string(f1)+" ")
elseif (e3*b2)<0 then
    f1=((e2*b2)-(b1*e3))/(b2-e3)
    disp("second approximation: "+string(f1)+" ")
elseif (e3*a2)<0 then
    f1=((e2*a2)-(a1*e3))/(a2-e3)
    disp("second approximation: "+string(f1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input second approximation or x2 that is rounded-off to 3 decimal places including its zeroes.")
f2=input ("x2 or second approximation: ")
disp("value of x2: "+string(f2)+" ")
if f2==e2 then
    disp (" "+string(f2)+" is the root.")
    abort
elseif f2==d2 then
    disp (" "+string(f2)+" is the root.")
    abort
else
    horner(y,[f2])
end
f3=horner(y,[f2]) // f3=f(x2)
disp("value of f(x2): "+string(f3)+" ")
if f3==0 then
    disp (" "+string(f2)+" is the root.")
    abort
elseif f3~=0 then
    disp("Please input f(x2)that is rounded-off to 3 decimal places including its zeroes.")
    f3=input ("value of f(x2): ")
end
if (f3*e3)<0 then
    g1=((f2*e3)-(e2*f3))/(e3-f3)
    disp("third approximation: "+string(g1)+" ")
elseif (f3*d3)<0 then
    g1=((f2*d3)-(d2*f3))/(d3-f3)
    disp("third approximation: "+string(g1)+" ")
elseif (f3*b2)<0 then
    g1=((f2*b2)-(b1*f3))/(b2-f3)
    disp("third approximation: "+string(g1)+" ")
elseif (f3*a2)<0 then
    g1=((f2*a2)-(a1*f3))/(a2-f3)
    disp("third approximation: "+string(g1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")

```

```

disp("Contact Email: bernardoraevon@gmail.com")
abort
end
disp("Please input third approximation or x3 that is rounded-off to 3 decimal places including its zeroes.")
g2=input ("x3 or third approximation: ")
disp("value of x3: "+string(g2)+" ")
if g2==f2 then
    disp (" "+string(g2)+" is the root.")
    abort
elseif g2==e2 then
    disp (" "+string(g2)+" is the root.")
    abort
elseif g2==d2 then
    disp (" "+string(g2)+" is the root.")
    abort
else
    horner(y,[g2])
end
g3=horner(y,[g2]) // g3=f(x3)
disp("value of f(x3): "+string(g3)+" ")
if g3==0 then
    disp (" "+string(g2)+" is the root.")
    abort
elseif g3~=0 then
    disp("Please input f(x3)that is rounded-off to 3 decimal places including its zeroes.")
    g3=input ("value of f(x3): ")
end
if (g3*f3)<0 then
    h1=((g2*f3)-(f2*g3))/(f3-g3)
    disp("fourth approximation: "+string(h1)+" ")
elseif (g3*e3)<0 then
    h1=((g2*e3)-(e2*g3))/(e3-g3)
    disp("fourth approximation: "+string(h1)+" ")
elseif (g3*d3)<0 then
    h1=((g2*d3)-(d2*g3))/(d3-g3)
    disp("fourth approximation: "+string(h1)+" ")
elseif (g3*b2)<0 then
    h1=((g2*b2)-(b1*g3))/(b2-g3)
    disp("fourth approximation: "+string(h1)+" ")
elseif (g3*a2)<0 then
    h1=((g2*b2)-(a1*g3))/(a2-g3)
    disp("fourth approximation: "+string(h1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input fourth approximation or x4 that is rounded-off to 3 decimal places including its zeroes.")
h2=input ("x4 or fourth approximation: ")
disp("value of x4: "+string(h2)+" ")
if h2==g2 then
    disp (" "+string(h2)+" is the root.")
    abort
elseif h2==f2 then

```

```

    disp (""+string(h2)+" is the root.")
    abort
elseif h2==e2 then
    disp (""+string(h2)+" is the root.")
    abort
elseif h2==d2 then
    disp (""+string(h2)+" is the root.")
    abort
else
    horner(y,[h2])
end
h3=horner(y,[h2]) // h3=f(x4)
disp("value of f(x4): "+string(h3)+" ")
if h3==0 then
    disp (""+string(h2)+" is the root.")
    abort
elseif h3~=0 then
    disp("Please input f(x4)that is rounded-off to 3 decimal places including its zeroes.")
    h3=input ("value of f(x4): ")
end
if (h3*g3)<0 then
    i1=((h2*g3)-(g2*h3))/(g3-h3)
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*f3)<0 then
    i1=((h2*f3)-(f2*h3))/(f3-h3)
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*e3)<0 then
    i1=((h2*e3)-(e2*h3))/(e3-h3)
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*d3)<0 then
    i1=((h2*d3)-(d2*h3))/(d3-h3)
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*b2)<0 then
    i1=((h2*b2)-(b1*h3))/(b2-h3)
    disp("fifth approximation: "+string(i1)+" ")
elseif (h3*a2)<0 then
    i1=((h2*a2)-(a1*h3))/(a2-h3)
    disp("fifth approximation: "+string(i1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input fifth approximation or x5 that is rounded-off to 3 decimal places including its zeroes.")
i2=input ("x5 or fifth approximation: ")
disp("value of x5: "+string(i2)+" ")
if i2==h2 then
    disp (""+string(i2)+" is the root.")
    abort
elseif i2==g2 then
    disp (""+string(i2)+" is the root.")
    abort
elseif i2==f2 then
    disp (""+string(i2)+" is the root.")
    abort

```

```

elseif i2==e2 then
    disp (""+string(i2)+" is the root.")
    abort
elseif i2==d2 then
    disp (""+string(i2)+" is the root.")
    abort
else
    horner(y,[i2])
end
i3=horner(y,[i2]) // i3=f(x5)
disp("value of f(x5): "+string(i3)+" ")
if i3==0 then
    disp (""+string(i2)+" is the root.")
    abort
elseif i3~=0 then
    disp("Please input f(x5)that is rounded-off to 3 decimal places including its zeroes.")
    i3=input ("value of f(x5): ")
end
if (i3*h3)<0 then
    j1=((i2*h3)-(h2*i3))/(h3-i3)
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*g3)<0 then
    j1=((i2*g3)-(g2*i3))/(g3-i3)
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*f3)<0 then
    j1=((i2*f3)-(f2*i3))/(f3-i3)
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*e3)<0 then
    j1=((i2*e3)-(e2*i3))/(e3-i3)
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*d3)<0 then
    j1=((i2*d3)-(d2*i3))/(d3-i3)
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*b2)<0 then
    j1=((i2*b2)-(b1*i3))/(b2-i3)
    disp("sixth approximation: "+string(j1)+" ")
elseif (i3*a2)<0 then
    j1=((i2*a2)-(a1*i3))/(a2-i3)
    disp("sixth approximation: "+string(j1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input sixth approximation or x6 that is rounded-off to 3 decimal places including its zeroes.")
j2=input ("x6 or sixth approximation: ")
disp("value of x6: "+string(j2)+" ")
if j2==i2 then
    disp (""+string(j2)+" is the root.")
    abort
elseif j2==h2 then
    disp (""+string(j2)+" is the root.")
    abort
elseif j2==g2 then
    disp (""+string(j2)+" is the root.")

```

```

    abort
elseif j2==f2 then
    disp (" "+string(j2)+" is the root.")
    abort
elseif j2==e2 then
    disp (" "+string(j2)+" is the root.")
    abort
elseif j2==d2 then
    disp (" "+string(j2)+" is the root.")
    abort
else
    horner(y,[j2])
end
j3=horner(y,[j2]) // j3=f(x6)
disp("value of f(x6): "+string(j3)+" ")
if j3==0 then
    disp (" "+string(j2)+" is the root.")
    abort
elseif j3~=0 then
    disp("Please input f(x6)that is rounded-off to 3 decimal places including its zeroes.")
    j3=input ("value of f(x6): ")
end
if (j3*i3)<0 then
    k1=((j2*i3)-(i2*j3))/(i3-j3)
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*h3)<0 then
    k1=((j2*h3)-(h2*j3))/(h3-j3)
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*g3)<0 then
    k1=((j2*g3)-(g2*j3))/(g3-j3)
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*f3)<0 then
    k1=((j2*f3)-(f2*j3))/(f3-j3)
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*e3)<0 then
    k1=((j2*e3)-(e2*j3))/(e3-j3)
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*d3)<0 then
    k1=((j2*d3)-(d2*j3))/(d3-j3)
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*b2)<0 then
    k1=((j2*b2)-(b1*j3))/(b2-j3)
    disp("seventh approximation: "+string(k1)+" ")
elseif (j3*a2)<0 then
    k1=((j2*a2)-(a1*j3))/(a2-j3)
    disp("seventh approximation: "+string(k1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input seventh approximation or x7 that is rounded-off to 3 decimal places including its zeroes.")
k2=input ("x7 or seventh approximation: ")
disp("value of x7: "+string(k2)+" ")

```



```

if k2==j2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==i2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==h2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==g2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==f2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==e2 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k2==d2 then
    disp (" "+string(k2)+" is the root.")
    abort
else
    horner(y,[k2])
end
k3=horner(y,[k2]) // k3=f(x7)
disp("value of f(x7): "+string(k3)+" ")
if k3==0 then
    disp (" "+string(k2)+" is the root.")
    abort
elseif k3~=0 then
    disp("Please input f(x7)that is rounded-off to 3 decimal places including its zeroes.")
    k3=input ("value of f(x7): ")
end
if (k3*j3)<0 then
    l1=((k2*j3)-(j2*k3))/(j3-k3)
    disp("eigth approximation: "+string(l1)+" ")
elseif (k3*i3)<0 then
    l1=((k2*i3)-(i2*k3))/(i3-k3)
    disp("eigth approximation: "+string(l1)+" ")
elseif (k3*h3)<0 then
    l1=((k2*h3)-(h2*k3))/(h3-k3)
    disp("eigth approximation: "+string(l1)+" ")
elseif (k3*g3)<0 then
    l1=((k2*g3)-(g2*k3))/(g3-k3)
    disp("eigth approximation: "+string(l1)+" ")
elseif (k3*f3)<0 then
    l1=((k2*f3)-(f2*k3))/(f3-k3)
    disp("eigth approximation: "+string(l1)+" ")
elseif (k3*e3)<0 then
    l1=((k2*e3)-(e2*k3))/(e3-k3)
    disp("eigth approximation: "+string(l1)+" ")
elseif (k3*d3)<0 then
    l1=((k2*d3)-(d2*k3))/(d3-k3)
    disp("eigth approximation: "+string(l1)+" ")
elseif (k3*b2)<0 then
    l1=((k2*b2)-(b1*k3))/(b2-k3)

```

```

    disp("eighth approximation: "+string(l1)+" ")
elseif (k3*a2)<0 then
    l1=((k2*a2)-(a1*k3))/(a2-k3)
    disp("eighth approximation: "+string(l1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input eighth approximation or x8 that is rounded-off to 3 decimal places including its zeroes.")
l2=input ("x8 or eighth approximation: ")
disp("value of x8: "+string(l2)+" ")
if l2==k2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==j2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==i2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==h2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==g2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==f2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==e2 then
    disp (" "+string(l2)+" is the root.")
    abort
elseif l2==d2 then
    disp (" "+string(l2)+" is the root.")
    abort
else
    horner(y,[l2])
end
l3=horner(y,[l2]) // l3=f(x8)
disp("value of f(x8): "+string(l3)+" ")
if l3==0 then
    disp(" "+string(l2)+" is the root.")
    abort
elseif l3~=0 then
    disp("Please input f(x8)that is rounded-off to 3 decimal places including its zeroes.")
    l3=input ("value of f(x8): ")
end
if (l3*k3)<0 then
    m1=((l2*k3)-(k2*l3))/(k3-l3)
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*j3)<0 then
    m1=((l2*j3)-(j2*l3))/(j3-l3)
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*i3)<0 then

```

```

    m1=((l2*i3)-(i2*l3))/(i3-l3)
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*h3)<0 then
    m1=((l2*h3)-(h2*l3))/(h3-l3)
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*g3)<0 then
    m1=((l2*g3)-(g2*l3))/(g3-l3)
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*f3)<0 then
    m1=((l2*f3)-(f2*l3))/(f3-l3)
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*e3)<0 then
    m1=((l2*e3)-(e2*l3))/(e3-l3)
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*d3)<0 then
    m1=((l2*d3)-(d2*l3))/(d3-l3)
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*b2)<0 then
    m1=((l2*b2)-(b1*l3))/(b2-l3)
    disp("ninth approximation: "+string(m1)+" ")
elseif (l3*a2)<0 then
    m1=((l2*a2)-(a1*l3))/(a2-l3)
    disp("ninth approximation: "+string(m1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input ninth approximation or x9 that is rounded-off to 3 decimal places including its zeroes.")
m2=input ("x9 or ninth approximation: ")
disp("value of x9: "+string(m2)+" ")
if m2==l2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==k2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==j2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==i2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==h2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==g2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==f2 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m2==e2 then
    disp (" "+string(m2)+" is the root.")
    abort

```

```

elseif m2==d2 then
    disp (" "+string(m2)+" is the root.")
    abort
else
    horner(y,[m2])
end
m3=horner(y,[m2]) // m3=f(x9)
disp("value of f(x9): "+string(m3)+" ")
if m3==0 then
    disp (" "+string(m2)+" is the root.")
    abort
elseif m3~=0 then
    disp("Please input f(x9)that is rounded-off to 3 decimal places including its zeroes.")
    m3=input ("value of f(x9): ")
end
if (m3*l3)<0 then
    n1=((m2*l3)-(l2*m3))/(l3-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*k3)<0 then
    n1=((m2*k3)-(k2*m3))/(k3-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*j3)<0 then
    n1=((m2*j3)-(j2*m3))/(j3-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*i3)<0 then
    n1=((m2*i3)-(i2*m3))/(i3-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*h3)<0 then
    n1=((m2*h3)-(h2*m3))/(h3-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*g3)<0 then
    n1=((m2*g3)-(g2*m3))/(g3-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*f3)<0 then
    n1=((m2*f3)-(f2*m3))/(f3-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*e3)<0 then
    n1=((m2*e3)-(e2*m3))/(e3-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*d3)<0 then
    n1=((m2*d3)-(d2*m3))/(d3-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*b2)<0 then
    n1=((m2*b2)-(b1*m3))/(b2-m3)
    disp("tenth approximation: "+string(n1)+" ")
elseif (m3*a2)<0 then
    n1=((m2*a2)-(a1*m3))/(a2-m3)
    disp("tenth approximation: "+string(n1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input tenth approximation or x10 that is rounded-off to 3 decimal places including its zeroes.")

```

```

n2=input ("x10 or tenth approximation: ")
disp("value of x10: "+string(n2)+" ")
if n2==m2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==l2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==k2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==j2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==i2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==h2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==g2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==f2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==e2 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n2==d2 then
    disp (" "+string(n2)+" is the root.")
    abort
else
    horner(y,[n2])
end
n3=horner(y,[n2]) // n3=f(x10)
disp("value of f(x10): "+string(n3)+" ")
if n3==0 then
    disp (" "+string(n2)+" is the root.")
    abort
elseif n3~=0 then
    disp("Please input f(x10)that is rounded-off to 3 decimal places including its zeroes.")
    n3=input ("value of f(x10): ")
end
if (n3*m3)<0 then
    o1=((n2*m3)-(m2*n3))/(m3-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*l3)<0 then
    o1=((n2*l3)-(l2*n3))/(l3-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*k3)<0 then
    o1=((n2*k3)-(k2*n3))/(k3-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*j3)<0 then
    o1=((n2*j3)-(j2*n3))/(j3-n3)
    disp("eleventh approximation: "+string(o1)+" ")

```

```

elseif (n3*i3)<0 then
    o1=((n2*i3)-(i2*n3))/(i3-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*h3)<0 then
    o1=((n2*h3)-(h2*n3))/(h3-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*g3)<0 then
    o1=((n2*g3)-(g2*n3))/(g3-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*f3)<0 then
    o1=((n2*f3)-(f2*n3))/(f3-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*e3)<0 then
    o1=((n2*e3)-(e2*n3))/(e3-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*d3)<0 then
    o1=((n2*d3)-(d2*n3))/(d3-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*b2)<0 then
    o1=((n2*b2)-(b1*n3))/(b2-n3)
    disp("eleventh approximation: "+string(o1)+" ")
elseif (n3*a2)<0 then
    o1=((n2*a2)-(a1*n3))/(a2-n3)
    disp("eleventh approximation: "+string(o1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input eleventh approximation or x11 that is rounded-off to 3 decimal places including its zeroes.")
o2=input("x11 or eleventh approximation: ")
disp("value of x11: "+string(o2)+" ")
if o2==n2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==m2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==l2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==k2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==j2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==i2 then
    disp(" "+string(n2)+" is the root.")
    abort
elseif o2==h2 then
    disp(" "+string(o2)+" is the root.")
    abort
elseif o2==g2 then

```

```

    disp (""+string(o2)+" is the root.")
    abort
elseif o2==f2 then
    disp (""+string(o2)+" is the root.")
    abort
elseif o2==e2 then
    disp (""+string(o2)+" is the root.")
    abort
elseif o2==d2 then
    disp (""+string(o2)+" is the root.")
    abort
else
    horner(y,[o2])
end
o3=horner(y,[o2]) // o3=f(x11)
disp("value of f(x11): "+string(o3)+" ")
if o3==0 then
    disp (""+string(o2)+" is the root.")
    abort
elseif o3~=0 then
    disp("Please input f(x11)that is rounded-off to 3 decimal places including its zeroes.")
    o3=input ("value of f(x11): ")
end
if (o3*n3)<0 then
    p1=((o2*n3)-(n2*o3))/(n3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*m3)<0 then
    p1=((o2*m3)-(m2*o3))/(m3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*l3)<0 then
    p1=((o2*l3)-(l2*o3))/(l3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*k3)<0 then
    p1=((o2*k3)-(k2*o3))/(k3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*j3)<0 then
    p1=((o2*j3)-(j2*o3))/(j3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*i3)<0 then
    p1=((o2*i3)-(i2*o3))/(i3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*h3)<0 then
    p1=((o2*h3)-(h2*o3))/(h3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*g3)<0 then
    p1=((o2*g3)-(g2*o3))/(g3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*f3)<0 then
    p1=((o2*f3)-(f2*o3))/(f3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*e3)<0 then
    p1=((o2*e3)-(e2*o3))/(e3-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*d3)<0 then
    p1=((o2*d3)-(d2*o3))/(d3-o3)
    disp("twelfth approximation: "+string(p1)+" ")

```

```

elseif (o3*b2)<0 then
    p1=((o2*b2)-(b1*o3))/(b2-o3)
    disp("twelfth approximation: "+string(p1)+" ")
elseif (o3*a2)<0 then
    p1=((o2*a2)-(a1*o3))/(a2-o3)
    disp("twelfth approximation: "+string(p1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input twelfth approximation or x12 that is rounded-off to 3 decimal places including its zeroes.")
p2=input("x12 or twelfth approximation: ")
disp("value of x12: "+string(p2)+" ")
if p2==o2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==n2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==m2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==l2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==k2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==j2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==i2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==h2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==g2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==f2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==e2 then
    disp(" "+string(p2)+" is the root.")
    abort
elseif p2==d2 then
    disp(" "+string(p2)+" is the root.")
    abort
else
    horner(y,[p2])
end
p3=horner(y,[p2]) // p3=f(x12)

```



```

disp("value of f(x12): "+string(p3)+" ")
if p3==0 then
    disp (" "+string(p2)+" is the root.")
    abort
elseif p3~=0 then
    disp("Please input f(x12)that is rounded-off to 3 decimal places including its zeroes.")
    p3=input ("value of f(x12): ")
end
if (p3*o3)<0 then
    q1=((p2*o3)-(o2*p3))/(o3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*n3)<0 then
    q1=((p2*n3)-(n2*p3))/(n3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*m3)<0 then
    q1=((p2*m3)-(m2*p3))/(m3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*l3)<0 then
    q1=((p2*l3)-(l2*p3))/(l3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*k3)<0 then
    q1=((p2*k3)-(k2*p3))/(k3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*j3)<0 then
    q1=((p2*j3)-(j2*p3))/(j3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*i3)<0 then
    q1=((p2*i3)-(i2*p3))/(i3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*h3)<0 then
    q1=((p2*h3)-(h2*p3))/(h3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*g3)<0 then
    q1=((p2*g3)-(g2*p3))/(g3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*f3)<0 then
    q1=((p2*f3)-(f2*p3))/(f3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*e3)<0 then
    q1=((p2*e3)-(e2*p3))/(e3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*d3)<0 then
    q1=((p2*d3)-(d2*p3))/(d3-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*b2)<0 then
    q1=((p2*b2)-(b1*p3))/(b2-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
elseif (p3*a2)<0 then
    q1=((p2*a2)-(a1*p3))/(a2-p3)
    disp("thirteenth approximation: "+string(q1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end

```

```

disp("Please input thirteenth approximation or x13 that is rounded-off to 3 decimal places including its zeroes.")
q2=input ("x13 or thirteenth approximation: ")
disp("value of x13: "+string(q2)+" ")
if q2==p2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==o2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==n2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==m2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==l2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==k2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==j2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==i2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==h2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==g2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==f2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==e2 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q2==d2 then
    disp (""+string(q2)+" is the root.")
    abort
else
    horner(y,[q2])
end
q3=horner(y,[q2]) // q3=f(x13)
disp("value of f(x13): "+string(q3)+" ")
if q3==0 then
    disp (""+string(q2)+" is the root.")
    abort
elseif q3~=0 then
    disp("Please input f(x13)that is rounded-off to 3 decimal places including its zeroes.")
    q3=input ("value of f(x13): ")
end
if (q3*p3)<0 then

```

```

r1=((q2*p3)-(p2*q3))/(p3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*o3)<0 then
r1=((q2*o3)-(o2*q3))/(o3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*n3)<0 then
r1=((q2*n3)-(n2*q3))/(n3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*m3)<0 then
r1=((q2*m3)-(m2*q3))/(m3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*l3)<0 then
r1=((q2*l3)-(l2*q3))/(l3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*k3)<0 then
r1=((q2*k3)-(k2*q3))/(k3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*j3)<0 then
r1=((q2*j3)-(j2*q3))/(j3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*i3)<0 then
r1=((q2*i3)-(i2*q3))/(i3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*h3)<0 then
r1=((q2*h3)-(h2*q3))/(h3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*g3)<0 then
r1=((q2*g3)-(g2*q3))/(g3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*f3)<0 then
r1=((q2*f3)-(f2*q3))/(f3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*e3)<0 then
r1=((q2*e3)-(e2*q3))/(e3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*d3)<0 then
r1=((q2*d3)-(d2*q3))/(d3-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*b2)<0 then
r1=((q2*b2)-(b1*q3))/(b2-q3)
disp("fourteenth approximation: "+string(r1)+" ")
elseif (q3*a2)<0 then
r1=((q2*a2)-(a1*q3))/(a2-q3)
disp("fourteenth approximation: "+string(r1)+" ")
else
disp("ERROR.")
disp("Please notify Head Developer of error encountered. Thank you!")
disp("Contact Email: bernardoraevon@gmail.com")
abort
end
disp("Please input fourteenth approximation or x14 that is rounded-off to 3 decimal places including its zeroes.")
r2=input ("x14 or fourteenth approximation: ")
disp("value of x14: "+string(r2)+" ")
if r2==q2 then
disp (" "+string(r2)+" is the root.")

```

```

    abort
elseif r2==p2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==o2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==n2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==m2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==l2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==k2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==j2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==i2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==h2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==g2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==f2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==e2 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r2==d2 then
    disp (" "+string(r2)+" is the root.")
    abort
else
    horner(y,[r2])
end

r3=horner(y,[r2]) // r3=f(x14)
disp("value of f(x14): "+string(r3)+" ")
if r3==0 then
    disp (" "+string(r2)+" is the root.")
    abort
elseif r3~=0 then
    disp("Please input f(x14)that is rounded-off to 3 decimal places including its zeroes.")
    r3=input ("value of f(x14): ")
end
if (r3*q3)<0 then
    s1=((r2*q3)-(q2*r3))/(q3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")

```

```

elseif (r3*p3)<0 then
    s1=((r2*p3)-(p2*r3))/(p3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*o3)<0 then
    s1=((r2*o3)-(o2*r3))/(o3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*n3)<0 then
    s1=((r2*n3)-(n2*r3))/(n3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*m3)<0 then
    s1=((r2*m3)-(m2*r3))/(m3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*l3)<0 then
    s1=((r2*l3)-(l2*r3))/(l3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*k3)<0 then
    s1=((r2*k3)-(k2*r3))/(k3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*j3)<0 then
    s1=((r2*j3)-(j2*r3))/(j3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*i3)<0 then
    s1=((r2*i3)-(i2*r3))/(i3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*h3)<0 then
    s1=((r2*h3)-(h2*r3))/(h3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*g3)<0 then
    s1=((r2*g3)-(g2*r3))/(g3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*f3)<0 then
    s1=((r2*f3)-(f2*r3))/(f3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*e3)<0 then
    s1=((r2*e3)-(e2*r3))/(e3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*d3)<0 then
    s1=((r2*d3)-(d2*r3))/(d3-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*b2)<0 then
    s1=((r2*b2)-(b1*r3))/(b2-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
elseif (r3*a2)<0 then
    s1=((r2*a2)-(a1*r3))/(a2-r3)
    disp("fifteenth approximation: "+string(s1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input fifteenth approximation or x15 that is rounded-off to 3 decimal places including its zeroes.")
s2=input("x15 or fifteenth approximation: ")
disp("value of x15: "+string(s2)+" ")
if s2==r2 then

```

```

    disp (""+string(s2)+" is the root.")
    abort
elseif s2==q2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==p2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==o2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==n2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==m2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==l2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==k2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==j2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==i2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==h2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==g2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==f2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==e2 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s2==d2 then
    disp (""+string(s2)+" is the root.")
    abort
else
    horner(y,[s2])
end
s3=horner(y,[s2]) // s3=f(x15)
disp("value of f(x15): "+string(s3)+" ")
if s3==0 then
    disp (""+string(s2)+" is the root.")
    abort
elseif s3~=0 then
    disp("Please input f(x15)that is rounded-off to 3 decimal places including its zeroes.")
    s3=input ("value of f(x15): ")
end

```

```

if (s3*r3)<0 then
    t1=((s2*r3)-(r2*s3))/(r3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*q3)<0 then
    t1=((s2*q3)-(q2*s3))/(q3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*p3)<0 then
    t1=((s2*p3)-(p2*s3))/(p3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*o3)<0 then
    t1=((s2*o3)-(o2*s3))/(o3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*n3)<0 then
    t1=((s2*n3)-(n2*s3))/(n3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*m3)<0 then
    t1=((s2*m3)-(m2*s3))/(m3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*l3)<0 then
    t1=((s2*l3)-(l2*s3))/(l3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*k3)<0 then
    t1=((s2*k3)-(k2*s3))/(k3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*j3)<0 then
    t1=((s2*j3)-(j2*s3))/(j3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*i3)<0 then
    t1=((s2*i3)-(i2*s3))/(i3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*h3)<0 then
    t1=((s2*h3)-(h2*s3))/(h3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*g3)<0 then
    t1=((s2*g3)-(g2*s3))/(g3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*f3)<0 then
    t1=((s2*f3)-(f2*s3))/(f3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*e3)<0 then
    t1=((s2*e3)-(e2*s3))/(e3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*d3)<0 then
    t1=((s2*d3)-(d2*s3))/(d3-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*b2)<0 then
    t1=((s2*b2)-(b1*s3))/(b2-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
elseif (s3*a2)<0 then
    t1=((s2*a2)-(a1*s3))/(a2-s3)
    disp("sixteenth approximation: "+string(t1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort

```

```

end
disp("Please input sixteenth approximation or x16 that is rounded-off to 3 decimal places including its zeroes.")
t2=input("x16 or sixteenth approximation: ")
disp("value of x16: "+string(t2)+" ")
if t2==s2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==r2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==q2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==p2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==o2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==n2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==m2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==l2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==k2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==j2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==i2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==h2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==g2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==f2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==e2 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t2==d2 then
    disp (" "+string(t2)+" is the root.")
    abort
else
    horner(y,[t2])
end

```



```

t3=horner(y,[t2]) // t3=f(x16)
disp("value of f(x16): "+string(t3)+" ")
if t3==0 then
    disp (" "+string(t2)+" is the root.")
    abort
elseif t3~=0 then
    disp("Please input f(x16)that is rounded-off to 3 decimal places including its zeroes.")
    t3=input ("value of f(x16): ")
end
if (t3*s3)<0 then
    u1=((t2*s3)-(s2*t3))/(s3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*r3)<0 then
    u1=((t2*r3)-(r2*t3))/(r3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*q3)<0 then
    u1=((t2*q3)-(q2*t3))/(q3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*p3)<0 then
    u1=((t2*p3)-(p2*t3))/(p3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*o3)<0 then
    u1=((t2*o3)-(o2*t3))/(o3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*n3)<0 then
    u1=((t2*n3)-(n2*t3))/(n3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*m3)<0 then
    u1=((t2*m3)-(m2*t3))/(m3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*l3)<0 then
    u1=((t2*l3)-(l2*t3))/(l3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*k3)<0 then
    u1=((t2*k3)-(k2*t3))/(k3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*j3)<0 then
    u1=((t2*j3)-(j2*t3))/(j3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*i3)<0 then
    u1=((t2*i3)-(i2*t3))/(i3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*h3)<0 then
    u1=((t2*h3)-(h2*t3))/(h3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*g3)<0 then
    u1=((t2*g3)-(g2*t3))/(g3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*f3)<0 then
    u1=((t2*f3)-(f2*t3))/(f3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*e3)<0 then
    u1=((t2*e3)-(e2*t3))/(e3-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*d3)<0 then
    u1=((t2*d3)-(d2*t3))/(d3-t3)

```

```

    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*b2)<0 then
    u1=((t2*b2)-(b1*t3))/(b2-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
elseif (t3*a2)<0 then
    u1=((t2*a2)-(a1*t3))/(a2-t3)
    disp("seventeenth approximation: "+string(u1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input seventeenth approximation or x17 that is rounded-off to 3 decimal places including its zeroes.")
u2=input("x17 or seventeenth approximation: ")
disp("value of x17: "+string(u2)+" ")
if u2==t2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==s2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==r2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==q2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==p2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==o2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==n2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==m2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==l2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==k2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==j2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==i2 then
    disp(" "+string(u2)+" is the root.")
    abort
elseif u2==h2 then
    disp(" "+string(u2)+" is the root.")
    abort

```

```

elseif u2==g2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==f2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==e2 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u2==d2 then
    disp (" "+string(u2)+" is the root.")
    abort
else
    horner(y,[u2])
end
u3=horner(y,[u2]) // u3=f(x17)
disp("value of f(x17): "+string(u3)+" ")
if u3==0 then
    disp (" "+string(u2)+" is the root.")
    abort
elseif u3~=0 then
    disp("Please input f(x17)that is rounded-off to 3 decimal places including its zeroes.")
    u3=input ("value of f(x17): ")
end
if (u3*t3)<0 then
    v1=((u2*t3)-(t2*u3))/(t3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*s3)<0 then
    v1=((u2*s3)-(s2*u3))/(s3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*r3)<0 then
    v1=((u2*r3)-(r2*u3))/(r3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*q3)<0 then
    v1=((u2*q3)-(q2*u3))/(q3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*p3)<0 then
    v1=((u2*p3)-(p2*u3))/(p3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*o3)<0 then
    v1=((u2*o3)-(o2*u3))/(o3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*n3)<0 then
    v1=((u2*n3)-(n2*u3))/(n3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*m3)<0 then
    v1=((u2*m3)-(m2*u3))/(m3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*l3)<0 then
    v1=((u2*l3)-(l2*u3))/(l3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*k3)<0 then
    v1=((u2*k3)-(k2*u3))/(k3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*j3)<0 then
    v1=((u2*j3)-(j2*u3))/(j3-u3)

```

```

    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*i3)<0 then
    v1=((u2*i3)-(i2*u3))/(i3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*h3)<0 then
    v1=((u2*h3)-(h2*u3))/(h3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*g3)<0 then
    v1=((u2*g3)-(g2*u3))/(g3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*f3)<0 then
    v1=((u2*f3)-(f2*u3))/(f3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*e3)<0 then
    v1=((u2*e3)-(e2*u3))/(e3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*d3)<0 then
    v1=((u2*d3)-(d2*u3))/(d3-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*b2)<0 then
    v1=((u2*b2)-(b1*u3))/(b2-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
elseif (u3*a2)<0 then
    v1=((u2*a2)-(a1*u3))/(a2-u3)
    disp("eighteenth approximation: "+string(v1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input eighteenth approximation or x18 that is rounded-off to 3 decimal places including its zeroes.")
v2=input("x18 or eighteenth approximation: ")
disp("value of x18: "+string(v2)+" ")
if v2==u2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==t2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==s2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==r2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==q2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==p2 then
    disp(" "+string(v2)+" is the root.")
    abort
elseif v2==o2 then
    disp(" "+string(v2)+" is the root.")
    abort

```

```

elseif v2==n2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==m2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==l2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==k2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==j2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==i2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==h2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==g2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==f2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==e2 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v2==d2 then
    disp (" "+string(v2)+" is the root.")
    abort
else
    horner(y,[v2])
end
v3=horner(y,[v2]) // v3=f(x18)
disp("value of f(x18): "+string(v3)+" ")
if v3==0 then
    disp (" "+string(v2)+" is the root.")
    abort
elseif v3~=0 then
    disp("Please input f(x18)that is rounded-off to 3 decimal places including its zeroes.")
    v3=input ("value of f(x18): ")
end
if (v3*u3)<0 then
    w1=((v2*u3)-(u2*v3))/(u3-v3)
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*t3)<0 then
    w1=((v2*t3)-(t2*v3))/(t3-v3)
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*s3)<0 then
    w1=((v2*s3)-(s2*v3))/(s3-v3)
    disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*r3)<0 then
    w1=((v2*r3)-(r2*v3))/(r3-v3)

```

```

disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*q3)<0 then
w1=((v2*q3)-(q2*v3))/(q3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*p3)<0 then
w1=((v2*p3)-(p2*v3))/(p3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*o3)<0 then
w1=((v2*o3)-(o2*v3))/(o3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*n3)<0 then
w1=((v2*n3)-(n2*v3))/(n3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*m3)<0 then
w1=((v2*m3)-(m2*v3))/(m3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*l3)<0 then
w1=((v2*l3)-(l2*v3))/(l3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*k3)<0 then
w1=((v2*k3)-(k2*v3))/(k3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*j3)<0 then
w1=((v2*j3)-(j2*v3))/(j3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*i3)<0 then
w1=((v2*i3)-(i2*v3))/(i3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*h3)<0 then
w1=((v2*h3)-(h2*v3))/(h3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*g3)<0 then
w1=((v2*g3)-(g2*v3))/(g3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*f3)<0 then
w1=((v2*f3)-(f2*v3))/(f3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*e3)<0 then
w1=((v2*e3)-(e2*v3))/(e3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*d3)<0 then
w1=((v2*d3)-(d2*v3))/(d3-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*b2)<0 then
w1=((v2*b2)-(b1*v3))/(b2-v3)
disp("nineteenth approximation: "+string(w1)+" ")
elseif (v3*a2)<0 then
w1=((v2*a2)-(a1*v3))/(a2-v3)
disp("nineteenth approximation: "+string(w1)+" ")
else
disp("ERROR.")
disp("Please notify Head Developer of error encountered. Thank you!")
disp("Contact Email: bernardoraevon@gmail.com")
abort
end

```

```

disp("Please input nineteenth approximation or x19 that is rounded-off to 3 decimal places including its zeroes.")
w2=input("x19 or nineteenth approximation: ")
disp("value of x19: "+string(w2)+" ")
if w2==v2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==u2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==t2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==s2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==r2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==q2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==p2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==o2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==n2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==m2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==l2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==k2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==j2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==i2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==h2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==g2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==f2 then
    disp (" "+string(w2)+" is the root.")
    abort
elseif w2==e2 then

```

```

disp (" "+string(w2)+" is the root.")
abort
elseif w2==d2 then
disp (" "+string(w2)+" is the root.")
abort
else
horner(y,[w2])
end
w3=horner(y,[w2]) // w3=f(x19)
disp("value of f(x19): "+string(w3)+" ")
if w3==0 then
disp (" "+string(w2)+" is the root.")
abort
elseif w3~=0 then
disp("Please input f(x19)that is rounded-off to 3 decimal places including its zeroes.")
w3=input ("value of f(x19): ")
end
if (w3*v3)<0 then
z1=((w2*v3)-(v2*w3))/(v3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*u3)<0 then
z1=((w2*u3)-(u2*w3))/(u3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*t3)<0 then
z1=((w2*t3)-(t2*w3))/(t3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*s3)<0 then
z1=((w2*s3)-(s2*w3))/(s3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*r3)<0 then
z1=((w2*r3)-(r2*w3))/(r3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*q3)<0 then
z1=((w2*q3)-(q2*w3))/(q3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*p3)<0 then
z1=((w2*p3)-(p2*w3))/(p3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*o3)<0 then
z1=((w2*o3)-(o2*w3))/(o3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*n3)<0 then
z1=((w2*n3)-(n2*w3))/(n3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*m3)<0 then
z1=((w2*m3)-(m2*w3))/(m3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*l3)<0 then
z1=((w2*l3)-(l2*w3))/(l3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*k3)<0 then
z1=((w2*k3)-(k2*w3))/(k3-w3)
disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*j3)<0 then
z1=((w2*j3)-(j2*w3))/(j3-w3)
disp("twentieth approximation: "+string(z1)+" ")

```



```

elseif (w3*i3)<0 then
    z1=((w2*i3)-(i2*w3))/(i3-w3)
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*h3)<0 then
    z1=((w2*h3)-(h2*w3))/(h3-w3)
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*g3)<0 then
    z1=((w2*g3)-(g2*w3))/(g3-w3)
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*f3)<0 then
    z1=((w2*f3)-(f2*w3))/(f3-w3)
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*e3)<0 then
    z1=((w2*e3)-(e2*w3))/(e3-w3)
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*d3)<0 then
    z1=((w2*d3)-(d2*w3))/(d3-w3)
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*b2)<0 then
    z1=((w2*b2)-(b1*w3))/(b2-w3)
    disp("twentieth approximation: "+string(z1)+" ")
elseif (w3*a2)<0 then
    z1=((w2*a2)-(a1*w3))/(a2-w3)
    disp("twentieth approximation: "+string(z1)+" ")
else
    disp("ERROR.")
    disp("Please notify Head Developer of error encountered. Thank you!")
    disp("Contact Email: bernardoraevon@gmail.com")
    abort
end
disp("Please input twentieth approximation or x20 that is rounded-off to 3 decimal places including its zeroes.")
z2=input("x20 or twentieth approximation: ")
disp("value of x20: "+string(z2)+" ")
if z2==w2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==v2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==u2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==t2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==s2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==r2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==q2 then
    disp(" "+string(z2)+" is the root.")
    abort
elseif z2==p2 then

```

```

    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==o2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==n2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==m2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==l2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==k2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==j2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==i2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==h2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==g2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==f2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==e2 then
    disp (" "+string(z2)+" is the root.")
    abort
elseif z2==d2 then
    disp (" "+string(z2)+" is the root.")
    abort
else
    horner(y,[z2])
end
z3=horner(y,[z2]) // z3=f(x20)
disp("value of f(x20): "+string(z3)+" ")
if z3==0 then
    disp (" "+string(z2)+" is the root.")
    abort
else
    disp("ERROR: Exceeded the capabilities of the program!")
    abort
end

```

III. Program Instructions

How to Use G6-WFPMRA 1.0.2:

Input polynomial function in this format:

`poly([a,b,c,-->nth],%x%,%coeff%)`

Wherein: Letters a,b,c,-->nth (excluding x) corresponds to the coefficients of your desired polynomial function in an ascending order in terms of degree.

Wherein: The symbol % should be replaced with quotation marks (**IMPORTANT**)

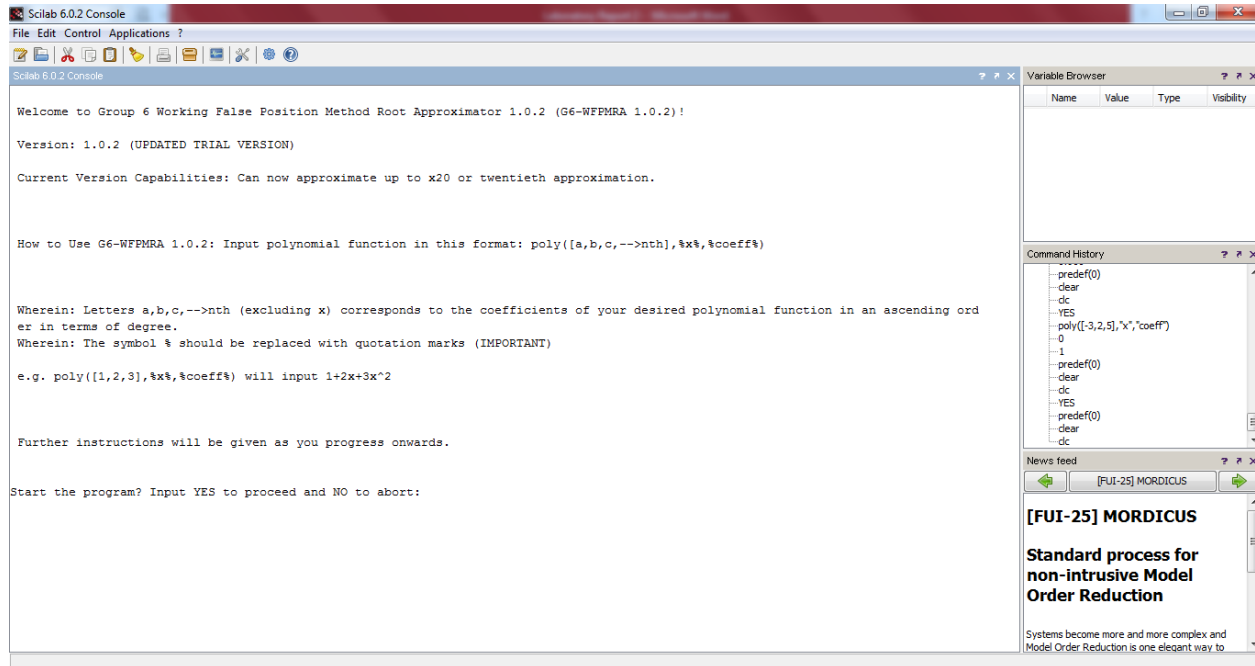
e.g. `poly([1,2,3],%x%,%coeff%)` will input **1+2x+3x²**

(Note: Due to the mechanics of Scilab, % is displayed in the instructions instead of quotation marks to avoid conflicts within the program codes.)

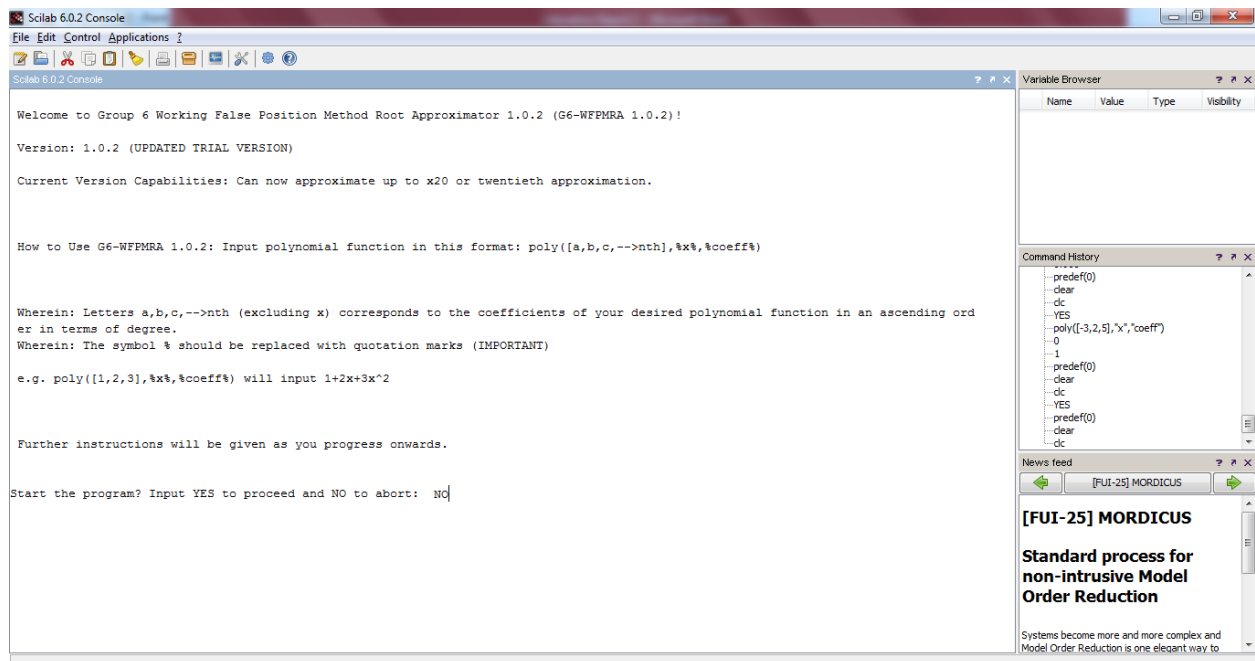
Further instructions will be given as you progress onwards.

IV. Sample Output

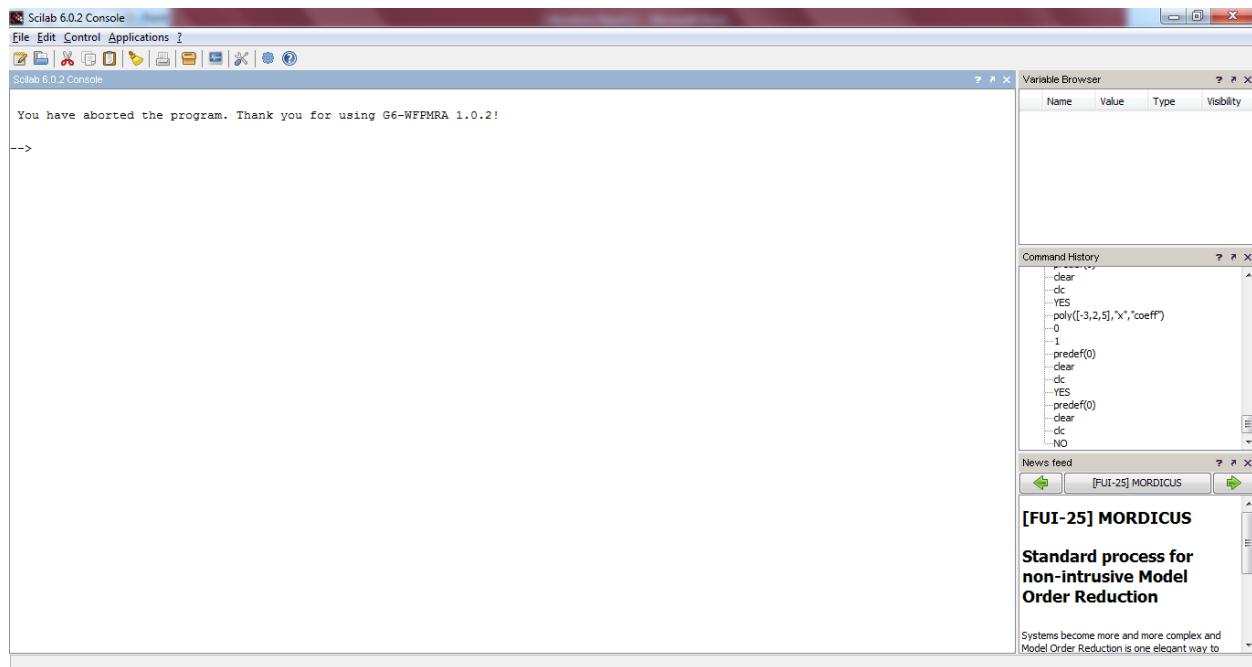
Program opened. Brief description and introduction of the program's current version as well as the set of instructions to be followed by the user will be displayed. The user will then be prompted for confirmation to either start or abort the program.



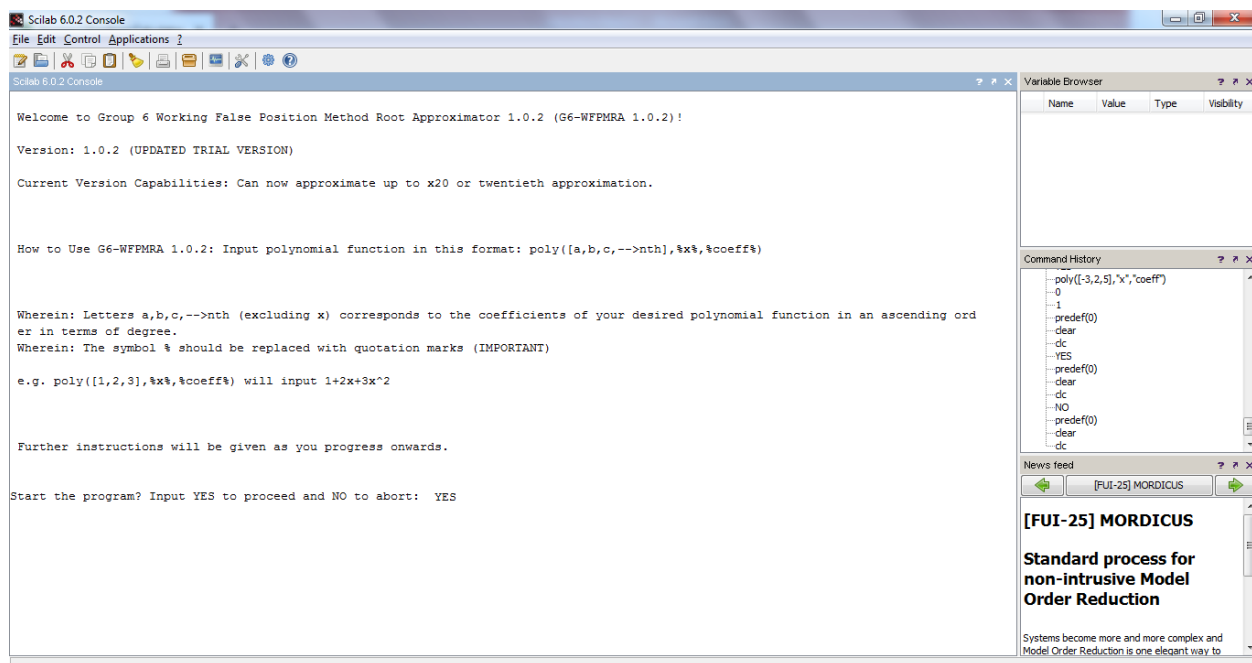
Scenario 1: User inputs "NO".



This is what will happen if the user's answer is "NO".



Scenario 2: User inputs "YES".



Scenario 1: After inputting “YES”, the user will be prompted to input desired polynomial function as well as the values of **a** and **b**. In this scenario, the algorithm has identified an error because the user had inputted a wrong assumption of lower and upper limits. As a result, the program will then prompt the user to input another different pair of values for the limits.

```

Scilab 6.0.2 Console
File Edit Control Applications ?
Wherein: The symbol % should be replaced with quotation marks (IMPORTANT)
e.g. poly([1,2,3],%x,%coeff%) will input 1+2x+3x^2

Further instructions will be given as you progress onwards.

Start the program? Input YES to proceed and NO to abort: YES

PROGRAM START!!

Input polynomial function:      poly([-3,2,5], "x", "coeff")
Input lower limit or value of a: 1
Input upper limit or value of b: 2

value of f(a):      4
value of f(b):      21
ERROR: f(a)*f(b) is not less than 0.

Please reinput lower and upper limits.

Input lower limit or value of a:

```

Scenario 2: After inputting “YES”, the user will be prompted to input desired polynomial function as well as the values of **a** and **b**. In this scenario, the user has inputted a polynomial function as well as a right assumption of lower and upper limits. As a result, the system will calculate for the initial approximation and will determine if it is a root. If so, the program will end. If not, it will prompt the user to round off its value as well as the value of **f(x₀)** to 3 decimal places and reinput the said values to start calculation for the next approximation.

```

Scilab 6.0.2 Console
File Edit Control Applications ?
Further instructions will be given as you progress onwards.

Start the program? Input YES to proceed and NO to abort: YES

PROGRAM START!!

Input polynomial function:      poly([-3,2,5], "x", "coeff")
Input lower limit or value of a: 0
Input upper limit or value of b: 1

value of f(a):      -3
value of f(b):      4

initial approximation:  0.4285714

Please input initial approximation or x0 that is rounded-off to 3 decimal places including its zeroes.
x0 or initial approximation:  0.429

value of x0:  0.429
value of f(x0):  -1.221795

Please input f(x0) that is rounded-off to 3 decimal places including its zeroes.
value of f(x0):  -1.222

```

After calculating the next approximation, the system will determine if it is a root. If not, the system will prompt the user to reinput the value of the approximation after rounding it off to 3 decimal places. This iterative process will continue until the root is determined. However, if the root is still not identified after the 20th approximation, the program will automatically abort as it is designed, in its current version, to only approximate until the 20th approximation.

```

Scilab 6.0.2 Console
File Edit Control Applications ?

PROGRAM START!!

Input polynomial function:      poly([-3,2,5],"x","coeff")
Input lower limit or value of a: 0
Input upper limit or value of b: 1

value of f(a):      -3
value of f(b):      4
initial approximation: 0.4285714

Please input initial approximation or x0 that is rounded-off to 3 decimal places including its zeroes.
x0 or initial approximation: 0.429

value of x0: 0.429
value of f(x0): -1.221795

Please input f(x0) that is rounded-off to 3 decimal places including its zeroes.
value of f(x0): -1.222

first approximation: 0.723847

Please input first approximation or x1 that is rounded-off to 3 decimal places including its zeroes.
x1 or first approximation: 0.724

```

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```

poly([-3,2,5],"x","coeff")
1
2
abort
predef(0)
clear
dc
YES
poly([-3,2,5],"x","coeff")
0
1
0.429
-1.222

```

News feed

Scilab 6.1.1 has been released!!!

Dear fellow users,

We have the pleasure to announce the release of the new version of Scilab. Check [here](#) to download and find more details about Scilab

```

Scilab 6.0.2 Console
File Edit Control Applications ?

value of x1: 0.724
value of f(x1): 1.06888

Please input f(x1) that is rounded-off to 3 decimal places including its zeroes.
value of f(x1): 1.069

second approximation: 0.5863505

Please input second approximation or x2 that is rounded-off to 3 decimal places including its zeroes.
x2 or second approximation: 0.586

value of x2: 0.586
value of f(x2): -0.11102

Please input f(x2) that is rounded-off to 3 decimal places including its zeroes.
value of f(x2): -0.111

third approximation: 0.5989814

Please input third approximation or x3 that is rounded-off to 3 decimal places including its zeroes.
x3 or third approximation: 0.599

value of x3: 0.599
value of f(x3): -0.007995

Please input f(x3) that is rounded-off to 3 decimal places including its zeroes.
value of f(x3): -0.008

```

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```

predef(0)
clear
dc
YES
poly([-3,2,5],"x","coeff")
0
1
0.429
-1.222
0.724
1.069
0.586
-0.111
0.599

```

News feed

Scilab 6.1.1 has been released!!!

Dear fellow users,

We have the pleasure to announce the release of the new version of Scilab. Check [here](#) to download and find more details about Scilab

Scilab 6.0.2 Console

File Edit Control Applications ?

Scilab 6.0.2 Console

```
value of f(x2): -0.11102

Please input f(x2) that is rounded-off to 3 decimal places including its zeroes.
value of f(x2): -0.111

third approximation: 0.5989814

Please input third approximation or x3 that is rounded-off to 3 decimal places including its zeroes.
x3 or third approximation: 0.599

value of x3: 0.599
value of f(x3): -0.007995

Please input f(x3) that is rounded-off to 3 decimal places including its zeroes.
value of f(x3): -0.008

fourth approximation: 0.5999285

Please input fourth approximation or x4 that is rounded-off to 3 decimal places including its zeroes.
x4 or fourth approximation: 0.600

value of x4: 0.6
value of f(x4): 0

0.6 is the root.

--> |
```

Variable Browser

Name	Value	Type	Visibility
------	-------	------	------------

Command History

```
--> dc
--> YES
--> poly([-3,2,5], "x", "coeff")
--> 0
--> 1
--> -0.429
--> -1.222
--> -0.724
--> 1.069
--> 0.586
--> -0.111
--> 0.599
--> -0.008
--> 0.600
```

News feed

Scilab 6.1.1 has been released!!!

Dear fellow users,

We have the pleasure to announce the release of the new version of Scilab. Check [here](#) to download and find more details about Scilab

V. Program Accuracy

Accuracy Test #1

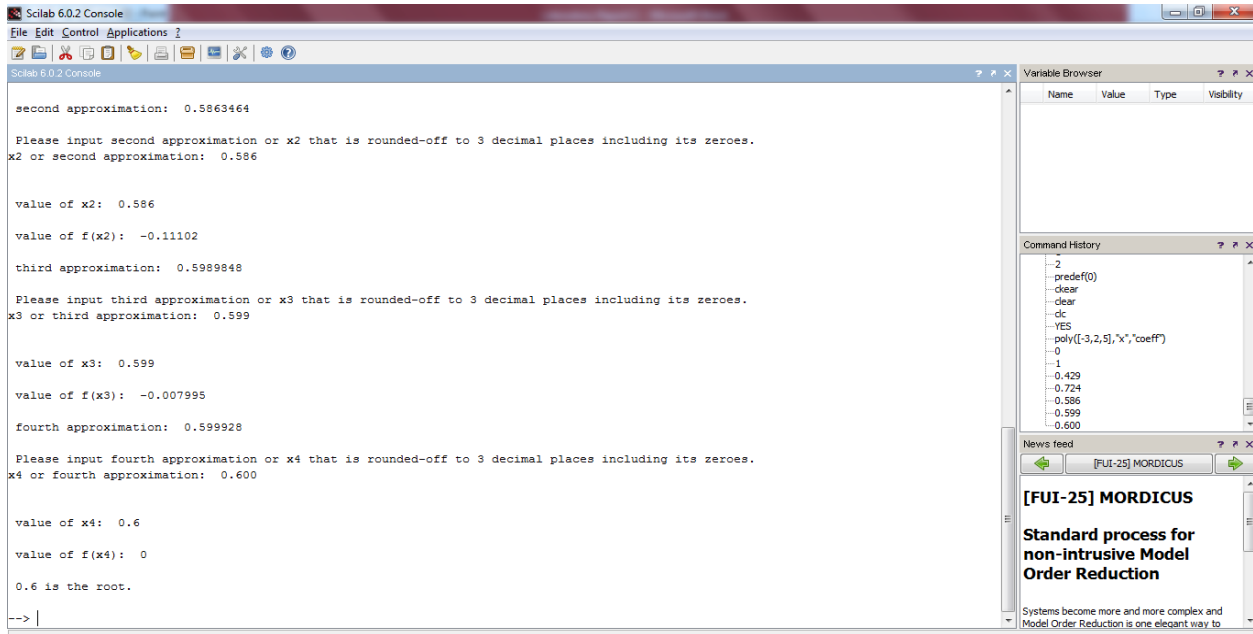
Polynomial function: $f(x) = 5x^2 + 2x - 3$

Lower limit or value of (a): 0

Upper limit or value of (b): 1

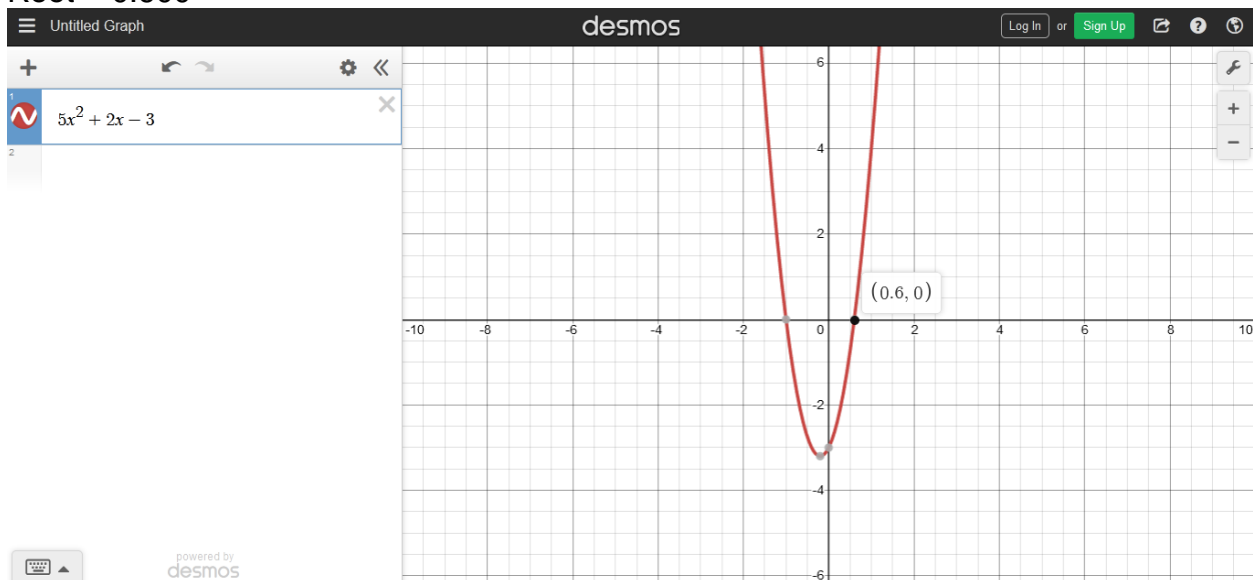
G6-WFPMRA 1.0.2 Results:

Root = 0.600



Desmos Graphing Calculator Results:

Root = 0.600



Accuracy Test #2

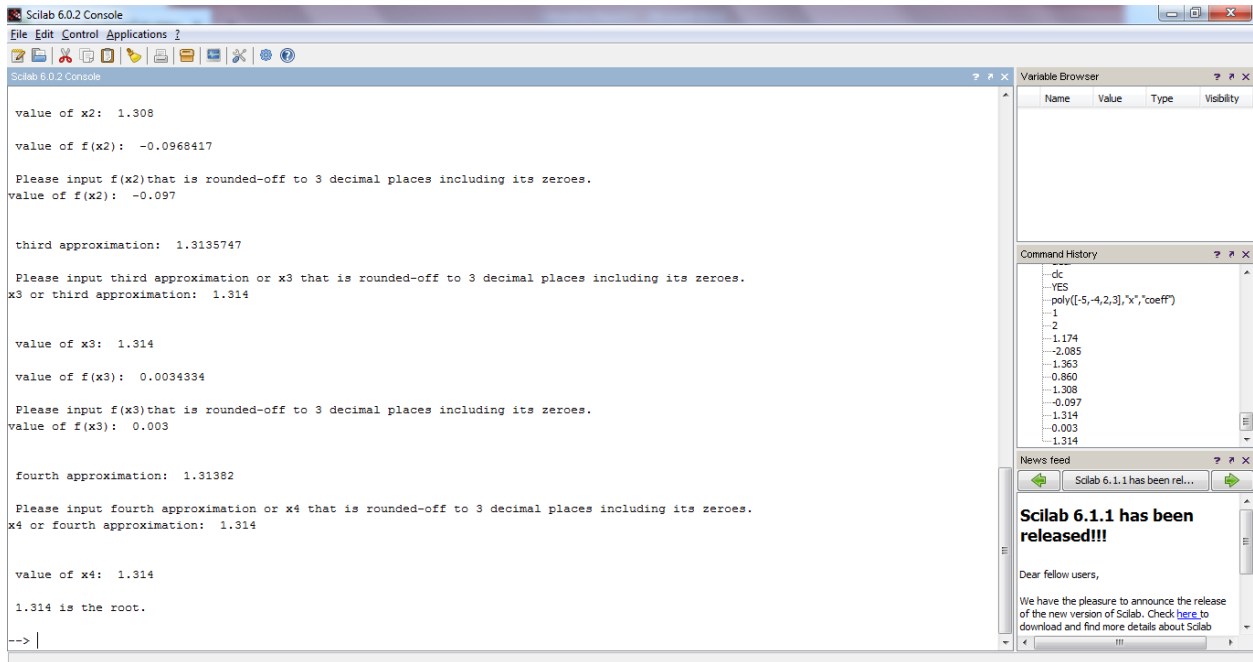
Polynomial function: $f(x) = 3x^3 + 2x^2 - 4x - 5$

Lower limit or value of (a): 0

Upper limit or value of (b): 1

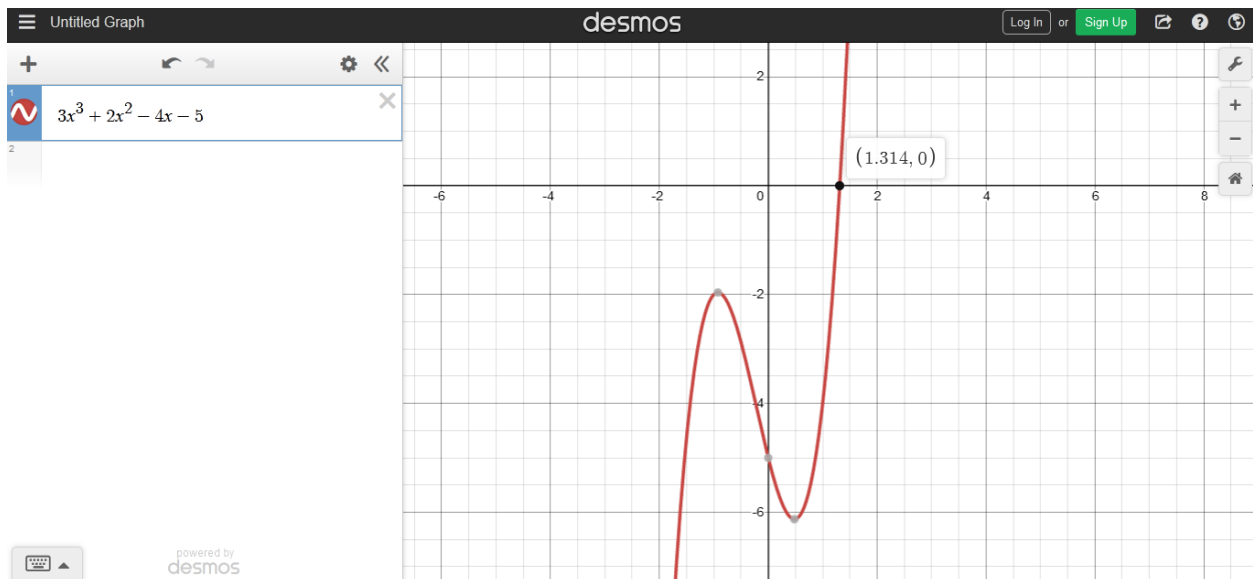
G6-WFPMRA 1.0.2 Results:

Root = 1.314



Desmos Graphing Calculator Results:

Root = 1.314



VI. Development Team Contributions

Development Team Members:

- ❖ Bernardo, Raevon Thaddeus C.
 - Head Developer & Programmer
 - Designed the algorithms of the working program
 - Final debugger of the program

- ❖ Bertumen, Charles Jefferson
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program

- ❖ Cabanes, Christine Joy P.
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program

- ❖ Cesar, John Lester M.
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program

- ❖ Landicho, Bhaves Nicolette D.
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program

- ❖ Solis, Johnloyd P.
 - Assistant Developer & Programmer
 - Assisted in conceptualizing the algorithms of the program
 - Assisted in assessing the performance of the trial version
 - Assisted in debugging the program

The development of the program was conducted systematically in order to maximize work efficiency, therefore, the final output was the result of total team effort and cooperation.

Head Developer's Remark